Protection des Accélérateurs Matériels de Cryptographie Symétrique

Sylvain Guilley

Université Paris VII

Protection des Accélérateurs Matériels de Cryptographie Symétrique

THÈSE D'HABILITATION

présentée pour l'obtention du Diplôme d'Habilitation à Diriger des Recherches de l'École normale supérieure (Spécialité Informatique)

par

Sylvain Guilley

soutenue publiquement le 14 décembre 2012 à TELECOM-ParisTech, devant le jury composé de

Claude Carlet ..... *Rapporteur*
Hervé Chabanne ..... *Rapporteur*
Arnand Durand ..... *Rapporteur*
Bart Preneel ..... *Rapporteur*
Jean-Luc Danger ..... *Examinateur*
Assia Tria ..... *Examinateur*
David Naccache ..... *Directeur, président du jury*

Travaux effectués au sein du groupe Systèmes Électroniques Numériques du Département Communications et Électroniques de TELECOM-ParisTech

à Charlotte

Contexte

Cette dernière décennie a vu se multiplier les appareils électroniques portatifs et les infrastructures de réseaux, dispositifs que l'on regroupe dans la catégorie des "systèmes embarqués". Leur sécurité est devenue particulièrement importante, au vu de la quantité et de la diversité des informations personnelles qu'ils traitent. Un écosystème scientifique riche s'est donc créé autour de la sécurité des systèmes embarqués : s'y côtoient académiques, industriels (aussi bien petites et moyennes entreprises que grands groupes) et gouvernementaux. En France, un panorama de cet écosystème a été effectué dans le cadre des journées sécurité du GdR SoC-SiP organisées par le LIRMM et le Lab-STICC. Aujourd'hui, cette communauté nationale est très active et collabore activement. Je suis d'ailleurs très fier de l'avoir représentée lors du dernier colloque CNRS Nippo-Français du JFFoE 2012 [167].

Ainsi, on constate de nombreux transferts, qui se concrétisent souvent par des démonstrateurs. Leur objectif est de valider qu'une idée de sécurisation est pertinente, y compris après le passage à l'implémentation. Effectivement, de nombreuses vulnérabilités sont introduites pendant les phases de raffinement et d'assemblage. Avant le développement d'un démonstrateur complet, on passe par différents prototypes, tournant sur des cibles technologiques intermédiaires tels que des processeurs généralistes ou des FPGA. En matière de prototypage, il y a eu également énormément de progrès. Une grande partie des travaux conduits dans le laboratoire de TELECOM-ParisTech peuvent aujourd'hui être évalués en émulation dans un FPGA en un mois de temps. Cette rapidité permet de tester incrémentalement de nouvelles idées : la recherche fondamentale rejoint les applications au sein du même laboratoire.

La sécurité des systèmes embarqués est par nature pluridisciplinaire. On trouvera donc dans ce rapport des parties touchant à la physique (signaux radio-fréquence, circuits R/L/C), l'électronique (modélisation des couplages de l'attaquant avec le dispositif), les statistiques (distingueurs, en présence de bruit), la théorie de l'information (évaluation de la vulnérabilité d'un système) et les mathématiques (codage et fonctions booléennes). J'ai en fait rencontré les disciplines mentionnées ci-dessus dans l'ordre chronologique suivant : partant d'une thèse de doctorat à très forte connotation « conception électronique », j'ai progressivement évolué vers la modélisation abstraite des failles et des exploits, sujet qui se rapporte davantage à des « sciences dures », même si l'objectif reste essentiellement appliqué. Par ailleurs, la production scientifique en sécurité des systèmes embarqués est elle-même très variée, traitant de générateurs d'aléa, de conception robuste, d'optimisation de contremesures, de codes, ou d'algèbre booléenne, pour ne citer que quelques sujets intéressant la communauté. Ceci se traduit en une multiplicité de collaborations, nationales et internationales, et une variété de supports de publication.

Le corollaire de cette diversité est le risque d'une segmentation scientifique, qui serait antinomique avec la volonté d'échanges pluridisciplinaires. C'est pourquoi le « faire-savoir » est particulièrement important. Il se met en place par de nombreuses tenues d'événements informels, de rencontres hors conférences officielles, ou de sessions spéciales de conférences. De plus, la pédagogie est primordiale, car avec l'accroissement du nombre de conférences, un résultat décrit dans un papier non réexpliqué par ailleurs a

peu de chances d'être lu. Ainsi, un objectif de ce rapport est aussi de faire le point sur une dizaine d'années de recherche, et de présenter les résultats obtenus de façon cohérente.

Remerciements

Les résultats de ce rapport n'auraient pas pu être obtenus sans le concours de nombreuses personnes. Je suis tout d'abord redevable à Renaud Pacalet, mon directeur de thèse de doctorat, de m'avoir lancé en 2002 sur la piste très porteuse de l'électronique sécurisée. Ensuite, je suis énormément reconnaissant à tous mes co-auteurs, avec qui nous avons raffiné de nombreuses idées, c'est-à-dire fait passer une intuition en une solution rigoureuse. Également, j'adresse mes sincères remerciements aux chercheurs doctorants qui travaillent actuellement leur thèse : il y a dans le laboratoire du 39 rue Dareau un gisement de savoir-faire exceptionnel, qui nous inspire tous mutuellement. Jean-Luc Danger a permis à ce laboratoire de monter de puissance, et je rends ici hommage à ses efforts continuels. Je souhaite de plus remercier Laurent Sauvage, pour sa curiosité scientifique et son engagement dans les réalisations, qui ont permis au laboratoire de s'organiser autour de plateformes pérennes. Enfin, merci au jury et notamment à mon directeur de HDR, David Naccache, pour ses encouragements et son caractère toujours volontaire, positif et exigeant.

Avant-propos

La plupart des résultats présentés dans ce manuscrit sont issus de publications évaluées par les pairs, et peuvent donc être considérés comme validés du point de vue scientifique. Ainsi, ce rapport n'apporte pas de résultats nouveaux par rapport à l'état de l'art, que mon travail a contribué à former. Néanmoins, on y trouvera une unification du contexte, présenté de façon pédagogique, et une articulation des différents travaux publiés à différents endroits avec différentes personnes. Il n'y a guère que la section qui traite de la résilience qui soit de nature plus exploratoire. Elle s'appuie sur quelques publications préliminaires de l'auteur. Néanmoins, une plus grande formalisation est certainement nécessaire pour fiabiliser les analyses présentées dans cette section. De plus, de nombreux points sont actuellement encore des sujets de recherche actifs. Les questions ouvertes sont évoquées dans les sous-parties de conclusions intermédiaires et consolidées dans la section de conclusions et perspectives. Enfin, il existe une très grande diversité de notations et de conventions divergentes dans la littérature scientifique. Nous proposons en annexe des notations claires et redémontrons tous les résultats utilisés dans ce manuscrit.

Ce document se structure en trois grande parties :

- **Chapitre 1 :** un exposé sur la "Protection des accélérateurs matériels de cryptographie symétrique",
- **Chapitre 2 :** un résumé du cursus et de la production scientifique de l'auteur, et
- **Chapitre 3 :** une série d'articles représentatifs du travail de l'auteur, sous leur version longue.

Table des matières

Chapitre 1

Protection des accélérateurs matériels de cryptographie symétrique

1.1 Introduction

La cryptographie est la science qui a pour objectif la sécurisation l'information. Les différents besoins élémentaires de sécurisation sont fournis par des algorithmes : par exemple, la confidentialité est assurée par le chiffrement. Le chiffrement de blocs de données est la primitive sur laquelle nous nous concentrerons par la suite. Beaucoup d'exemples ont été spécifiés et publiés. La plupart de ces algorithmes ont été analysés ouvertement par la communauté de recherche en cryptologie (selon le principe de Kerckhoffs [240, 241]), sans pourtant que des failles fatales ne soient trouvées. Certains sont donc aujourd'hui standardisés, et considérés essentiellement comme sûrs. Pour être précis, il faut mentionner que quelques-uns sont sujets à des attaques. Par exemple, dans le cas du DES [336], il existe une cryptanalyse dite différentielle [44] et une autre dite linéaire [298]. Ou bien, dans le cas de l'AES [337], une cryptanalyse a été annoncée récemment : il s'agit de l'analyse bicyclique [48]. Néanmoins, il faut relativiser la portée de ces cryptanalyses : elles permettent certes de retrouver la clé plus facilement qu'une recherche exhaustive, mais à un coût qui reste néanmoins prohibitif ($2^{37} < 2^{56}$ invocations pour simple-DES, et $2^{126.1} \lessapprox 2^{128}$ pour l'AES-128). Ainsi, nous considérons que ces attaques ne sont pas réalistes. D'ailleurs, l'industrie s'en accommode volontiers. Ceci est particulièrement vrai pour DES, qui a été remplacé dès le 25 octobre 1999 par sa variante triple-DES (dans FIPS PUB 46-3). Le manque de réalisme de ces cryptanalyses signifie que dans leur état de maturité, elles ne compromettent pas la sécurité des algorithmes concernés ; elles restent toutefois intéressantes dans la mesure où elles soulignent des vulnérabilités latentes des primitives cryptographiques. Avec davantage de recherches, ces vulnérabilités pourraient devenir exploitables, comme cela avait été le cas des failles sur la famille SHA [338] qui se sont raffinées jusqu'à devenir réalistes après des années d'amélioration.

Ceci étant dit, les attaques que nous étudions dans ce rapport ont une complexité maximale de 2^{32}, soit environ quelques milliards ($\approx 1.000.000.000$) de requêtes à la primitive cryptographique. Toutefois, pour attaquer en moins d'un milliard d'invocations au module de chiffrement par blocs, certaines conditions propices à l'attaquant doivent être réunies. Presque toujours, c'est l'aspect « concret » du dispositif qui est à la source des failles. Un premier cas peut survenir lorsque la fonctionnalité est erronée. Bref, une erreur fatale est « présente dans l'œuf ». Cela peut être une faille introduite volontairement : on parle alors de porte dérobée. Cela signifie qu'il existe un moyen très difficilement détectable par un utilisateur (*i.e.* en boîte noire) qui permet à un attaquant de faire sortir le secret. Cependant, un évaluateur, examinant le système en boîte blanche, pourra éventuellement facilement constater des branchements conditionnels suspicieux. Ou encore, la faille peut provenir d'une maladresse d'implémentation, comme un comportement non décrit ou mal documenté. Bref, tous ces problèmes sont communément appelés « *bugs* ». Dans certains cas, les vulnérabilités qu'introduisent les *bugs* peuvent être exploitées. Par exemple, la possibilité qu'a un attaquant de pouvoir, sur un même dispositif,

- effacer une clé DES par morceau et
- choisir la clé

permet de monter une attaque qui trouve la clé en $\#k/8 = 8$ étapes [18, §3.3]. Mais il existe des moyens de détecter ces problèmes (outils de vérification de code) et aussi des méthodes préventives (*e.g.* méthodes formelles) pour les éviter dès la phase de conception. Ainsi, nous supposerons par la suite que l'implémentation étudiée est correcte (ce qui est une hypothèse de travail nécessaire mais réaliste). Une autre attaque sur l'implémentation peut se faire par une analyse d'émanations physiques ou une perturbation via l'environnement. Effectivement, les algorithmes cryptographiques s'appuient implicitement sur des hypothèses, telles que le secret d'une clé ou le caractère unidistribué d'un générateur d'aléa. Dit autrement, il y a toujours tacitement un ancrage de la sécurité dans le matériel d'exécution. Or, physiquement parlant, il existe des méthodes pour accéder à des données enfouies ou pour influencer le dispositif de sorte qu'il calcule en-dehors de ses spécifications fonctionnelles [444]. Elles peuvent être classifiées en trois catégories, en fonction de leur invasivité [1].

- **Les attaques en observation** [250] consistent à enregistrer passivement un canal caché sur lequel fuit une version dégradée des données manipulées en interne au composant (cf. l'article d'encyclopédie sur la « *Cryptophthora* » dans [330]) ; elles sont décrites dans le cas des analyses de courant sur carte à puce dans le livre *"Power Analysis Attacks: Revealing the Secrets of Smart Cards"* [290], des systèmes à processeurs 32-bit dans le chapitre 8, intitulé *"Side-Channel Attacks on the Embedded System"* (pages 163-222) du livre *"Security in Embedded Devices"* [136] et dans le cas des FPGA au chapitre 3 intitulé *"Side Channel Attacks"* du livre *"Security Trends for FPGAS – From Secured to Secure Reconfigurable Systems"* [16].
- **Les attaques en perturbation** viennent fauter des données internes [329], dans

1. La sécurité des systèmes électroniques a aussi un pendant non-invasif, où l'adversaire utilise les voies des couches hautes (logicielles) pour réaliser ses exploits [146]. Nous n'aborderons pas ces aspects dans ce rapport.

l'objectif d'en déduire des informations sur leur valeur ou bien d'éliminer des hypothèses sur les secrets. Suivant les conditions expérimentales, il y a deux types d'attaques possibles : soit la perturbation est reproductible, ce qui permet de tester l'état d'une valeur en vérifiant si oui ou non la faute a eu un effet (analyse dite "*safe error*"), soit la perturbation est non maîtrisée, mais le résultat (*i.e.* le cryptogramme) est connu : il est alors possible d'éliminer des secrets par étude des différences en sortie (analyse dite "*differential fault analysis*" ou DFA) ; l'ouvrage de référence est *"Fault Analysis in Cryptography"* [233].

– **Les attaques en manipulation** viennent sonder le circuit ou le modifier (dans les deux cas, souvent en aveugle). Il y a peu d'ouvrage de référence ; on peut citer des livres qui traitent de la préparation et de la réparation des circuits, comme *"Integrated Circuit Failure Analysis: A Guide to Preparation Techniques"* [24].

L'objectif de la sécurité des systèmes embarqués est de mettre à mal ces attaques, mais tout en respectant symétriquement des aspects réalistes d'implémentation. Effectivement, si le système à protéger n'est pas accessible à l'attaquant, les attaques d'implémentation listées plus haut ne sont simplement pas applicables (car l'attaquant n'y a un accès physique ni en « lecture », ni en « écriture » [203]). Il est donc primordial de prendre en ligne de compte les contraintes liées au caractère « embarqué » de la plateforme. On peut notamment mentionner : vitesse de calcul, taille (quantité de ressources à mobiliser), consommation d'énergie, maintenabilité et prouvabilité du code. Néanmoins, dans un souci de lisibilité de l'approche, nous privilégierons toujours la sécurité sur la performance. Les compromis sécurité / performance seront aussi discutés, mais dans un cadre très rigoureux (*cf.* annexe G).

Comme annoncé, l'étude est plutôt orientée vers la cryptographie symétrique (comme dans la thèse de Johan Borst [51]), car c'est la partie qui est utilisée à flux tendu dans les protocoles pour les calculs intensifs. La cryptographie symétrique s'appuie sur des primitives de chiffrement par bloc [243] ; Les plus utilisés au niveau international sont DES, AES, Twofish, Serpent, mais il existe aussi des chiffrements nationaux, comme GOST en Union Soviétique, Camellia au Japon, SEED et ARIA en Corée, *etc.* Nous illustrerons nos résultats sur DES et AES, qui sont représentatifs des deux grands familles de chiffrements bloc, à savoir les réseaux de Feistel et les réseaux de permutations et substitution (dits SPN, pour « *substitution permutation networks* »). De plus, dans ce document, nous étudions les attaques et les contremesures (CM) propres au matériel, c'est-à-dire aux implémentations CMOS [472] câblées. Les attaques permettent de simplifier considérablement l'extraction de secret. Typiquement, une attaque va retrouver avec une probabilité de 90% la secret avec N interactions (mesure de canaux auxiliaires, injection de faute, *etc.*), où N est inférieur au milliard. Bien souvent, ce nombre est surévalué, car des contingences expérimentales de mise en œuvre imposent une grande redondance. Mais une fois l'attaque établie de façon fiable, il est possible de l'optimiser, ce qui fait baisser N de plusieurs ordres de grandeur [347, 416]. Par conséquent, on va s'attendre à ce qu'une CM soit très significativement efficace, et permette notamment de multiplier N par plusieurs ordres de grandeurs.

Par rapport au logiciel, le matériel a les spécificités suivantes :

– Il est aisé de réaliser du parallélisme lors du traitement.
– L'architecture est plus maîtrisée ; notamment, il est possible de garantir la simultanéité de traitement entre plusieurs données [2] ou l'équilibrage de certaines ressources dupliquées.

Aussi bien en ASIC qu'en FPGA, le concepteur dispose de différentes ressources pour réaliser une fonctionnalité « à façon ». Il pioche parmi :

– des portes logiques combinatoires, pour la « *glue* » ou pour les fonctionnalités *ad hoc*,
– des portes logiques séquentielles, comme les registres, pour les points de mémorisation temporaire,
– des mémoires, pour les grosses masses de données (caches ou stockage statique),
– éventuellement des blocs de propriété intellectuelle, tels que des opérateurs arithmétiques, comme des DSP (*Digital Signal Processors*).

On peut ainsi voir la conception matérielle comme un assemblage de briques élémentaires constituant un processeur sur-mesure.

Les contraintes imposées au matériel de cryptographie sont les suivantes :

– Il faut des contremesures qui soient robustes de façon unitaire. Ceci est légèrement différent des contremesures des cartes à puces, où l'empilement de contremesures est de règle. Car les cartes à puces sont des systèmes embarqués très particuliers, notamment très intégrés, où les CM sont dissimulées dans du silicium, Ainsi, des expédients comme des générateurs de bruit, ou du lissage de courant [326, 327, 464] sont autant de contremesures globales assez efficaces. Effectivement, dans les cartes à puces, il est difficile pour un attaquant de sonder spécifiquement une zone plutôt qu'une autre, et donc de s'affranchir des générateurs de bruit. Les cartographies électromagnétiques [398, 111] ont été réalisées sur des FPGAs, aux dimensions d'au moins un ordre de grandeur supérieur aux cartes à puces. Comme dans les FPGAs les contremesures analogiques ne sont pas possibles (l'utilisateur peut programmer la logique, mais pas la modifier). De plus, l'attaquant a tout loisir d'envoyer des entrées / sorties car le dispositif n'est pas borné à une ou peu d'applications. Mais la contrainte de sécurité unitaire aidera notre démarche : il est plus rigoureux de spécifier une contremesure qui est censée marcher au mieux sans compter sur d'autres entraves.
– De plus, il ne faudra que peu dégrader la vitesse. Ainsi, l'accent est mis dans ce rapport sur les contremesures rapides, *i.e.* dont le débit n'est que peu modifié (ralentissement d'un facteur au plus deux).

Le reste du manuscrit est organisé de la façon suivante. La section 1.2 donne un panorama des méthodes de conception des systèmes embarqués ; ceci est utile pour comprendre les moyens d'action d'un concepteur de système sécurisé. Ensuite, nous étudions deux types de contremesures au niveau de l'implémentation : le masquage et la dissimulation. L'effet de ces deux stratégies est d'amenuiser le lien entre les observations X (publiques, car « fuies » inexorablement) d'un canal caché et les données internes Y (privées, car

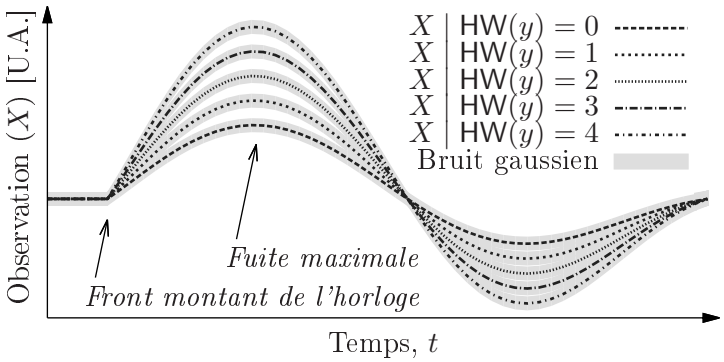
2. Le déséquilibre temporel pour l'échantillonnage des bascules D est appelé *skew*, et est typiquement de seulement quelques picosecondes dans les technologiques submicroniques profondes.

compromettant directement la clé secrète). Comme illustré dans la Fig. 1.1, le masquage cherche à rendre X indépendant de Y (tout en gardant X entropique), alors que la dissimulation cherche à rendre X constant. Ainsi, l'objectif du masquage (dit du premier ordre) est d'avoir une moyenne de X par classes de $Y = y$ égale pour toutes les valeurs de y, et de la dissimulation est d'avoir une valeur de X qui ne dépend simplement pas de Y. D'où une variance du bruit plus grande pour le masquage que pour la dissimulation. Effectivement, dans la CM de masquage, il s'ajoute au bruit ambiant un bruit de calcul (aussi appelé bruit algorithmique), qui n'existe pas dans le cas de la dissimulation. Notons également que l'amplitude des signaux est multipliée par deux dans les cas et d'un masquage du premier ordre et d'une dissimulation, car le taux de d'activité est doublé (soit en moyenne, soit de façon déterministe), eu égard à la duplication de matériel engendrée par l'une et l'autre de ces CMs. Les illustrations de la Fig. 1.1 montrent des captures temporelles d'une fuite : cela signifie que la fuite X est en réalité une fonction $X(t)$ du temps t. La fuite ne démarre véritablement qu'au début d'un calcul, ce qui correspond dans les systèmes séquentiels synchrones à un front montant de l'horloge. Par la suite, nous ne considérerons qu'une unique date, représentant typiquement celle où la fuite est la plus importante. Celle-ci a lieu quelques nanosecondes après le début du calcul, ce qui correspond au temps nécessaire pour que les portes logiques commutent en fonction de la valeur sensible. Effectivement, la forme d'onde est celle d'une sinusoïde amortie, car le couplage entre l'activité du circuit (rapide) et la sonde est imparfait, ce qui provoque des rebonds (plus lents) dans le signal $X(t)$. Ce filtrage lié à la transduction lors de la mesure étale l'information fuie dans le temps, ce qui n'est en pratique pas gênant, si la fréquence de coupure du filtrage n'est pas trop faible par rapport à la fréquence des commutations internes au circuit.

L'étude de ces deux contremesures est découpée en deux parties : leur principe (vision du défenseur) et leur évaluation (vision de l'attaquant). Ces aspects sont abordés de façon concise dans les articles [257, 199], qui fournissent un aperçu didactique du domaine. Ce manuscrit les développe amplement, en détaillant les constructions des contremesures et en raffinant les analyses de sécurité. Le principe du masquage est explicité dans la Sec. 1.3, et son évaluation dans la Sec. 1.4. De même, le principe de la dissimulation figure dans la Sec. 1.5 et son évaluation dans la Sec. 1.6. À côté des contre-mesures d'implémentation, on trouve des contremesures d'usage : pour que les attaques fonctionnent, un certain nombre d'hypothèses doivent être vérifiées. En les invalidant au niveau de l'appel des primitives cryptographiques, on peut ainsi protéger les secrets avec un travail nul ou minime sur l'implémentation des primitives cryptographiques. Cette stratégie, qui ne souffre pas de défauts de sécurité, est appelée résilience. Elle est abordée dans la section 1.7. Les conclusions et les perspectives se trouvent dans la Sec. 1.8. Enfin, les notations et certains résultats calculatoires sont relégués en annexe (Sec. 1.9).

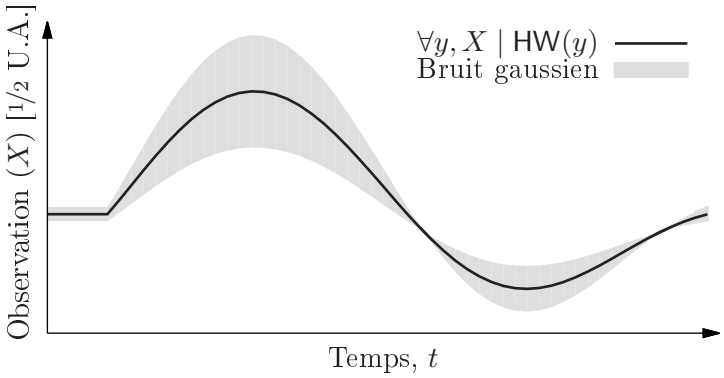
(a) Sans protection – fuite en poids de Hamming

Pas de CM : $X = \mathsf{HW}(y)$, for $\mathsf{HW}(y) \in [\![0, 4]\!]$



(b) Avec protection par masquage

Masquage : $X \perp\!\!\!\perp Y$ (*independence* dans le cas idéal)



(c) Avec protection par dissimulation

Dissimulation : X est constant (cas idéal)

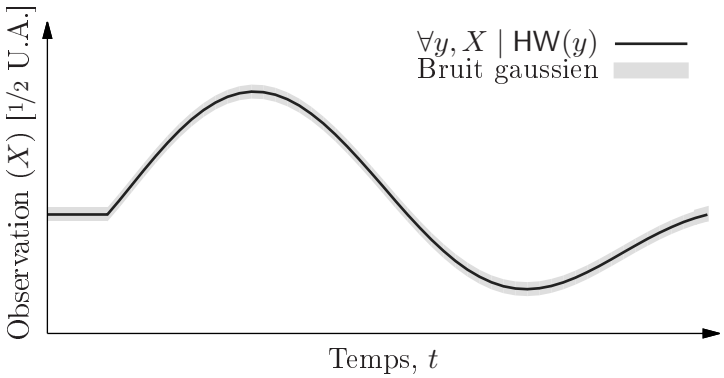


FIGURE 1.1 – Exemples d'observation X et de son lien en fonction des valeurs de $\mathsf{HW}(Y) \in [\![0, 4]\!]$ (car nous supposons ici que $Y \in \mathbb{F}_2^4$), dans les cas **(a)** non-protégé, **(b)** protégé par masquage, et **(c)** protégé par dissimulation.

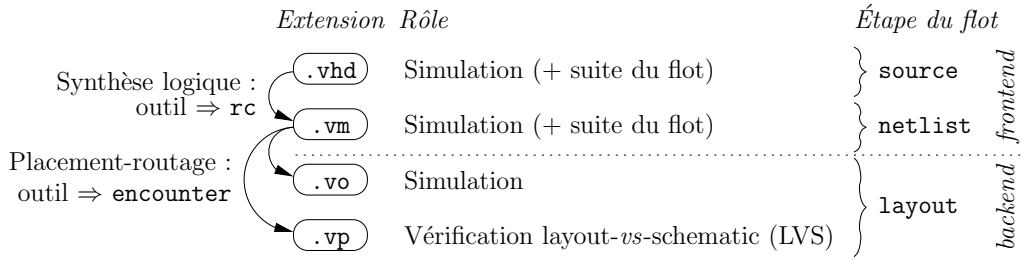


FIGURE 1.2 – Flot ASIC : du source aux masques.

1.2 Conception et modélisation d'un système embarqué sécurisé

1.2.1 Flot

La conception de matériel démarre par un code écrit dans un langage de description de matériel. Ce peut être typiquement VHDL [229] ou Verilog [228]. Ces langages permettent de décrire des processus parallèles synchronisés. La spécificité de ces langages est qu'ils permettent tout à la fois de spécifier formellement la fonctionnalité et d'être raffinables pour une projection technologique. Cela signifie d'une part qu'ils peuvent être simulés, dans des outils que l'on appelle « simulateurs logiques ». D'autre part, cela signifie qu'ils peuvent être synthétisés, c'est-à-dire traduits dans une description équivalente sur le plan fonctionnel mais faisant appel aux éléments de base offerts par la technologie ciblée. Avant cette étape dite de « synthèse logique », la description du matériel est *comportementale* (portable), alors qu'après, elle est *structurelle* (adaptée pour une technologie donnée). La vue structurelle est aussi appelée *netlist*, *i.e.* ensemble d'équipotentielles. Une dernière étape, dite de *backend*, est résumée sous le nom de placement-routage (ou P&R). La description structurelle, en terme de d'instances de portes, est placée sur la surface disponible et est interconnectée (ou routée). On aboutit, sur ASIC, au dessin des masques, ou *layout*, et sur FPGA, au ficher de configuration, ou *bitstream*. Un exemple de flot de conception est donné dans la Fig. 1.2 : les étapes de raffinement sur le source (`.vhd` → `.vm` → `.vo` → `.vp`) sont réalisées par les outils de la société CADENCE. Les différents fichiers sont le source (`.vhd`), la *netlist* après synthèse (`.vm`), la *netlist* après P&R, sans (`.vo`) ou avec (`.vp`) ports d'alimentation. Effectivement, le P&R concerne aussi bien les signaux de données que les nœuds globaux non présents dans le code source, comme typiquement les alimentations.

Les flots de conception pour ASIC (circuits cousus main) et FPGA (circuits reconfigurables) se ressemblent. La différence principale est que le concepteur peut tout redéfinir en ASIC (les portes logiques, le réseau de routage, les alimentations), alors qu'en FPGA, il n'a comme degrés de liberté que ceux laissés par le kit d'utilisation du FPGA. La finesse de réalisation est donc plus grande en ASIC qu'en FPGA. Il est possible de guider le placement pour la plupart des marques de FPGA. Cependant, le routage est plus contrô-

lable pour les FPGA de la marque Xilinx que ceux de la marque Altera ; par exemple, les copier-coller sont possibles avec Xilinx mais pas avec Altera (à cause d'une plus grande complexité du *layout*). Néanmoins, les FPGA présentent tout de même des avantages non-négligeables. D'une part ils sont moins coûteux pour des petites séries que les ASIC ; ceci est dû au coût d'entrée pour un ASIC que constitue la fabrication du jeu des masques. D'autre part, le cycle de développement d'un FPGA est bien plus rapide. Effectivement, il n'y a ni étape de fabrication (signalons qu'en technologie submicronique, un circuit passe plus d'un mois en usine) ni étape de vérification (c'est la chaîne de synthèse et de P&R qui garantit la conformité entre la programmation du FPGA et le code source).

Ainsi, on peut d'ores et déjà voir qu'il y a trois niveaux où un concepteur peut insérer et tester des CM dans son système :

1. Au niveau du code source, avec test par **simulation** logique.
2. Au niveau d'une *netlist*, avec test par **émulation** *in situ* dans un FPGA ; à ce niveau, on aussi tester approximativement des CM qui exigent du placement, voire du routage contraint. On peut certainement se dire que comme un FPGA est moins contrôlable est plus dissipatif qu'un ASIC, le résultat d'une évaluation émulée sera certainement moins bon que celui d'une évaluation sur la cible finale.
3. Au niveau des masques de fabrication, après avoir envoyé le circuit en fonderie. Au retour, on peut faire des tests grandeur nature sur le composant réel, grâce à des **mesures**.

1.2.2 Contremesures d'implémentation

Les contremesures peuvent s'immiscer à différentes étapes du flot de conception. Il est rare car dangereux d'insérer des contremesures dès le code source. Effectivement, il est nécessaire que la CM n'altère pas la fonctionnalité. Or les CM vont typiquement complexifier la description du système. Comme le code source est confié à un synthétiseur logique, qui a pour objectif d'implémenter au mieux la description avec les ressources disponibles, il y a de grandes chances qu'il simplifie, voire supprime complètement la CM. Les CM vont donc, en général, travailler au niveau de l'implémentation, dans notre cas au niveau *netlist*. Concrètement, on va s'attendre à ce que la CM modifie la durée de l'algorithme ou le nombre de ressources nécessaires à sa réalisation. À niveau de sécurité égal, on va préférer la CM qui minimise le temps d'exécution et la quantité de ressources dont elle a besoin. La suite de cette section détaille différentes stratégies de CM contre les différents types de menace.

Contre les attaques en observation, il existe deux options. Soit les données manipulées sont rendues aléatoires (ce que l'on appelle CM de masquage), soit elles sont rendues indiscernables (ce que l'on appelle CM de dissimulation). En terme de changement de valeurs, cela donne une CM où l'activité est non-prédictible et une autre où l'activité est constante.

Contre les attaques en perturbation, la détection des erreurs est l'approche conservatrice. Elle consiste à ne pas laisser sortir de résultat qui puisse être erroné. Néanmoins, si l'attaquant a toute latitude dans son injection, il sera en mesure de provoquer des

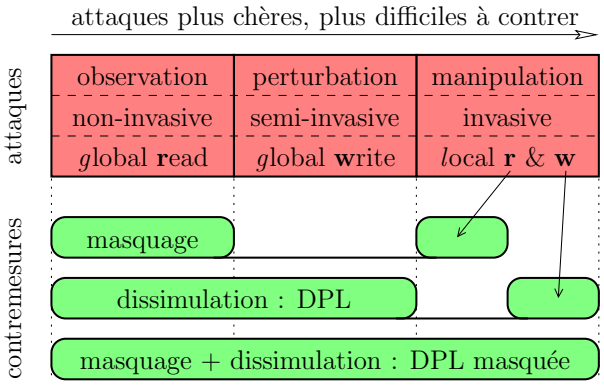


FIGURE 1.3 – Couverture des CM pour toutes les classes d'attaques physiques.

fautes qui ne seront pas détectées. La contremesure se caractérise donc par son taux de couverture. Il existe également une méthode plus audacieuse, qui consiste à laisser les fautes se propager, en ayant pris soin de vérifier au préalable que la CM assure qu'elles ne transportent pas d'information exploitable. Il se trouve que les CM qui implémentent la dissimulation vérifient aussi cette propriété intéressante. En revanche, les CM qui implémentent le masquage ne sont pas protégées contre les attaques en perturbation [52].

Les attaques en manipulation sont redoutablement puissantes. Certes le blindage du composant peut intimider un attaquant, mais sa couverture n'est pas complète. D'une part, il ne concerne que la face avant du composant. De plus, avec des outils avancés, il peut être partiellement défait [443]. Voilà pourquoi la protection naturelle est le masquage des données : si l'attaquant ne connaît pas la valeur des équipotentielles, il ne lui sera d'aucune utilité de les sonder [216]. Effectivement, la valeur relevée ne correspond pas à une variable sensible. Ici, la dissimulation n'a que peu d'intérêt, car dans la plupart de ces logiques, il existe un encodage de type vrai / faux. Ainsi, l'attaquant sondera-t-il des valeurs toujours vraies ou toujours inversées, mais ce de façon consistante. Bref, il ne perd qu'un seul bit d'entropie, ce que l'on ne peut pas objectivement qualifier de CM. Ceci dit, même avec du masquage, l'attaquant pourra injecter chirurgicalement des fautes. Pour une protection optimale, il est donc nécessaire de complémenter détection et masquage ou dissimulation et masquage [199]. Ceci est illustré dans la Fig. 1.3.

On constate que contre les attaques en observation, il faut soit un générateur d'aléa, soit une maîtrise de l'équilibrage. Certes l'usage d'un générateur d'aléa peut sembler simplifier la conception de la CM. Néanmoins, de façon tout à fait pragmatique, il s'agit d'un *deus ex machina*, dont il faut également prendre soin. Car cette ressource peut également fuir ou être fautée (par exemple forcée à une valeur constante ou faiblement entropique [296]). Ainsi, il n'y a pas trivialement de CM plus efficace qu'une autre : il faudra les examiner au cas par cas, selon le contexte.

Pour terminer, mentionnons les contremesures de résilience contre les attaques en injection de fautes [206]. L'idée est de ne laisser à l'attaquant ni le loisir de choisir où

tombe la faute ni de répéter un même calcul deux fois [90]. Sur ce second point, on voit qu'il ne s'agit pas d'une CM d'implémentation, mais d'une CM d'usage.

1.2.3 Méthodologie d'évaluation

Selon les contextes, le terme de méthodologie revêt différentes significations. Les évaluations standardisées, comme FIPS-140 [367] et les Critères Communs [1] (notés CC), comportent une phase documentaire de vérification de conformité et une phase expérimentale dite de test de pénétration. Si l'analyse documentaire est essentiellement réalisée en boîte blanche, les tests de pénétration ont plutôt lieu en boîte noire. Les modalités pour FIPS-140 sont détaillées dans le document annexe appelé "DTR" (*Derived Tests Requirements*), et pour les critères communs (version $\geqslant 3$) dans la classe `AVA_VLAN`. Il peut sembler paradoxal de ne pas utiliser les connaissances documentaires pour guider l'évaluation pratique. Mais il y a des raisons tout à fait valables à ce choix. D'une part, une contremesure bien décrite pourrait dissuader un évaluateur de même entreprendre quelque attaque que cela ne soit. Or, comme les protections œuvrent souvent au niveau de l'implémentation, il est tout à fait envisageable que le principe de la CM soit bon, mais que sa réalisation soit mauvaise. Ainsi, il est utile de tester sans trop se préoccuper de la connaissance des contremesures. D'autre part, il est aussi probable que la CM protège correctement de la menace contre laquelle elle a été conçue, mais pas de toutes les autres. Par exemple, une attaquant qui sait attaquer en manipulation contourne aisément des contremesures telles que la dissimulation. Or, l'objectif d'une évaluation n'est pas de tester la solidité d'une CM, mais du produit en entier.

Maintenant, ces évaluations visent à éprouver la résistance globale d'un système, via quelques tests choisis (philosophie FIPS) ou attaques choisies (philosophie CC). La conclusion de ces évaluations sera que le circuit n'a pas été attaqué avec succès par une équipe compétente avec un certain budget. Mais aucune autre extrapolation ne pourra être faite ; d'ailleurs, on affirme que la certification d'un produit est valide le jour de l'évaluation qui n'a pas mis en évidence de failles, mais périme dès le lendemain. Ainsi, cette façon de procéder ne permet pas de garantir une sécurité sur le long terme (ou *forward security*). Les évaluations, telles que considérées par les académiques, ont un autre objectif. Il s'agit de valider une CM, ou plutôt une abstraction de la CM, en termes de vulnérabilité et de sécurité. Par exemple, pour quantifier le risque d'une attaque sur les canaux auxiliaires [434], tout d'abord la quantité d'information fuie est mesurée (typiquement avec une métrique de théorie de l'information), puis la capacité à essuyer des attaques est quantifiée (typiquement avec une métrique dite de sécurité, comme un taux de succès). La première analyse sert à estimer le pire cas, *i.e.* la moindre dépendance entre les données sensibles et le canal caché. Maintenant, il se peut qu'il n'existe pas de distingueur qui sache transformer cette fuite (présente dans le canal caché) en avantage pour l'attaquant. Les raisons peuvent être de différentes natures :

- la fuite ne dépend pas de la clé, donc ne permet pas de construire un distingueur ;
- la fuite dépend de la clé, mais est injective dans la variable sensible, ce qui ne permet pas de construire "tel quel" un distingueur basé sur la théorie de l'information [364] ;

– pour utiliser la fuite dans l'objectif de retrouver un octet, il faut connaître tous les autres octets de la clé.

Pour être exhaustif, on peut formaliser également les étapes expérimentales de l'attaque, à savoir l'acquisition des mesures ou des fautes, leur prétraitement et leur exploitation. De telles instanciations de méthodologies ont été proposées et aident à rendre comparables des évaluations [424, 425]. La version 2 du "DPA contest" [446] poursuit le même objectif, en se restreignant toutefois à la phase d'exploitation.

Pourtant, pour être vraiment efficaces, les attaques passives nécessitent d'inférer une variable sensible. Il s'agit d'une valeur intermédiaire qui dépende d'un secret constant et qui soit calculable. Plus précisément, on s'attend à ce qu'elle soit prédictible moyennant des hypothèses en nombre gérable sur le secret. Typiquement, le nombre d'hypothèses est 2^6 pour DES, 2^8 pour AES, dans les cas de figure où l'algorithme est implémentée de façon découpée, *i.e.* en transposant directement la spécification en implémentation. Pour ces deux algorithmes, une stratégie « diviser-pour-règner » s'applique, ce qui permet de deviner la clé par morceaux de clé de tour. Dans certaines implémentations, il faut deviner jusqu'à 2^{32} variables intermédiaires, ce qui reste du domaine du faisable [321]. Toutefois la réelle difficulté est d'identifier les variables sensibles. Un exemple est donné dans [25], sans néanmoins détailler une méthodologie complète. Ainsi, la définition des modèles de fuite reste une activité essentiellement empirique, qui se décline au cas par cas. Effectivement, entre la connaissance exacte du modèle et un profilage exhaustif [69, 200], il y a un gouffre dans lequel l'évaluateur doit, grâce à son expertise, chercher en tâtonnant un modèle « plausible ». Notamment, les bons modèles de fuite dépendent de l'algorithme. Par exemple, pour l'AES, on peut deviner "$m \oplus k$" ou "$S(m \oplus k)$", où :

– m est un octet du message clair (ou *plaintext*),
– k est un octet de la clé de la première ronde (*i.e.* la clé maîtresse) et
– S est la boîte de substitution (abrégée en sbox) de l'AES, nommée `SubBytes`.

On peut aussi faire la même chose sur le dernier tour, avec "$c \oplus k$" ou "$S^{-1}(c \oplus k)$", où cette fois-ci :

– c est un octet du cryptogramme (ou *ciphertext*),
– k est un octet de la clé de la dernière ronde et
– S^{-1} est la fonction inverse de S (aussi appelée `InvSubBytes` dans le standard NIST/FIPS 197 qui décrit l'AES).

Certes, toutes ces variables seront obligatoirement calculées, donc on peut s'attendre à une dépendance entre elles et les traces mesurées si l'hypothèse sur la partie de clé k impliquée dans le calcul est correcte. Maintenant, si l'on connaît l'implémentation, on peut faire des modèles de fuite moins aveugles. Par exemple, sur le premier tour, il y a de la diffusion (`MixColumns`, `ShiftRows`) qui nous rend fastidieux la remontée d'un octet du premier tour (car il faudrait pour cela faire une hypothèse non pas sur un octet de clé k, mais sur une colonne de clé, soit 4 octets). Or si l'on connaît un état initial et un état final d'un registre, on peut appliquer une fonction de fuite "canonique", comme la distance de Hamming (*i.e.* le nombre de bits qui a changé). Ceci peut être fait sur le dernier tour de l'AES, car contrairement à la première ronde, il n'y a pas de diffusion qui mélange les bits dans les octets (en effet, `ShiftRows` est présent mais pas `MixColumns`). On peut donc

prédire l'état précédent d'un octet de l'état. Par exemple, pour l'octet à la position $(0,0)$, cela donne : "$S^{-1}(c \oplus k) \oplus c$", où "$S^{-1}(c \oplus k)$" est la valeur initiale de l'octet du registre et "c" est sa valeur finale. On vérifie simplement que "$S^{-1}(c \oplus k) \oplus c$" est une variable sensible : elle dépend d'un peu de clé (un octet, soit seulement 256 hypothèses) et elle est calculable connaissant le cryptogramme. Voilà encore deux points pour compléter cette brève introduction à « l'art de la définition de modèles de fuites » :

1. Pourquoi utiliser plutôt "$S^{-1}(c \oplus k)$" que "$c \oplus k$" ? C'est que quand on se trompe d'un bit sur l'hypothèse de clé k (*i.e.* l'hypothèse est correcte, sauf un bit), alors "$c \oplus k$" est aussi presque bon ; donc il sera dur de différentier la bonne clé des mauvaises clés, notamment celles qui sont proches en terme de distance de Hamming. On dit que la distance du distingueur correct à son plus proche rival [473] est faible. Alors qu'avec une fonction non-linéaire, comme "S^{-1}", qui sert justement à introduire de la confusion, si un bit est faux en entrée, alors en moyenne la moitié des bits en sortie sont affectés. Ceci est illustré dans la figure 1 de [25].
2. Quand on a affaire à un processeur matériel (aussi dit *hardware*), la distance de Hamming est la meilleure option, bien que les modèles en poids de Hamming marchent également (voir par exemple les cinq modèles étudiés dans [118, 200], et appliqués sur un même jeu de traces). Sur une implémentation logicielle (aussi dite *software*), c'est *a priori* plutôt un modèle en poids de Hamming qui est le plus pertinent.
   Dans un tel cas, l'attaque peut poser problème s'il existe des modèles de fuite identiques pour retrouver des parties différentes d'un secret. On peut penser par exemple à une attaque SCARE (*c.f.* Appendix I) qui chercherait une composante particulière d'une sbox. Effectivement toutes les composantes de la sbox s'expriment rigoureusement de la même façon, de telle sorte que la solution sera un mélange (très certainement *faux*) des solutions identifiées pour chaque bit. Cet effet de dégénerescence ne s'appliquerait pas dans le cas d'une « distance de Hamming », car la valeur connue (initiale ou finale) ferait intervenir explicitement une variable aléatoire indépendente, composante par composante.

1.3 Masquage : principe

Dans cette section sont étudiées les CM de masquage, qui sont des logiques à activité constante statistiquement.

1.3.1 Mélange d'aléa

Une variable aléatoire, appelée masque, s'ajoute au calcul. Comme il faut pouvoir réaliser l'opération inverse, le masquage s'appuiera sur une loi de groupe. Les masquages les plus usuels sont :

- Le masquage booléen [148] ;
- Le masquage multiplicatif [6] ;
- Le masquage affine [131] (la composée des deux précédents).

On peut voir le masquage comme un partage de secret probabiliste. L'intérêt du masquage Booléen est multiple. D'une part, le masquage est involutif, c'est-à-dire que l'opération de ou-exclusif sert et à masquer et à démasquer. D'autre part, l'ajout de la clé dans l'algorithme est transparent. Effectivement, si l'on note x la variable sensible, m le masque et k la clé, tous supposés avoir la même taille, alors on a $(x \oplus m) \oplus k = (x \oplus k) \oplus m$.

Nous allons donc détailler particulièrement le cas booléen.

1.3.2 Implémentations

1.3.2.1 Chemins de données parallèles

1.3.2.1.1 Cas le plus simple Le calcul est réalisé sur la donnée sensible masquée $x_m \doteq x \oplus m$ d'une part et sur le masque m d'autre part. Ces deux grandeurs sont appelées les deux parties ; elles vérifient bien que leur ou-exclusif renvoie la variable sensible démasquée x. Le passage des fonctions linéaires L peut se faire sur chaque partie indépendemment, car après passage par L, $L(x_m)$ et $L(m)$ satisfont toujours la contrainte que leur ou-exclusif est égal à $L(x)$. La traversée d'une fonction non-linéaire, comme une boîte de substitution S, est plus compliquée. Une solution consiste à introduire une fonction $S' : a, b \mapsto S'(a, b) \doteq \tilde{S}(a \oplus b) \oplus S(a)$, où $\tilde{S}$ est une sbox de même dimensions que S, et à calculer en parallèle :

$$\begin{cases} \tilde{S}(x_m) = \tilde{S}(x \oplus m) & // \text{ Chemin de la donnée masquée} \\ S'(x_m, m) = \tilde{S}(x_m \oplus m) \oplus S(x_m) = S(x) \oplus \tilde{S}(x \oplus m) & // \text{ Chemin du masque} \end{cases} \tag{1.1}$$

dont le ou-exclusif donnerait bien $S(x)$. Souvent, par commodité et éventuellement pour permettre du partage de ressources (souci d'économie), on choisit $\tilde{S} = S$. De plus, il y a apparition d'un nouveau masque, à savoir $\tilde{S}(x \oplus m) \oplus S(x)$. Étant données les bonnes propriétés différentielles de S, cette transformation de masque garantit effectivement une bonne indépendance entre le nouveau masque et l'ancien. Par ailleurs, on peut noter que la variable sensible démasquée apparaît dans l'Eqn. (1.1) ; cependant, le démasquage n'a pas réellement lieu, car l'opérateur S' est typiquement tabulé [413] (*i.e.* implémenté comme une mémoire). Néanmoins, il est bon de garder en tête qu'une décomposition (non « *white-box* » [403]), comme la logique USM (*Universal Sbox Masking* [274, Fig. 3]), serait fatale à la sécurité.

L'inconvénient d'un masquage tel que celui de l'Eqn. (1.1) est que la fonction S' a deux fois plus de bits d'adresses que S, et est donc très coûteuse. La figure 1.4 illustre comment ce masquage s'intègre dans un DES. Par rapport à une architecture de DES non protégée, représenté en noir, le matériel de masquage à ajouter est représenté en gris. Il est clair aussi bien dans les formules logiques du masquage que dans la figure que la partie ajoutée n'affecte pas le résultat, et qu'il faut donc bien veiller à ce qu'un outil de synthèse logique ne la supprime pas. Dans la figure 1.4, l'entrée supplémentaire k_i est la clé d'implémentation, *i.e.* le matériel de masquage. La clé k_c correspond à la clé cryptographique, en l'occurrence le résultat de $\mathrm{PC2}(\mathrm{CD}_i)$ pour chaque ronde $i \in [\![1, 16]\!]$.

La table S fait 2^6 mots de 4 bits (soit 256 bits) alors que la table S' fait 2^{12} mots de 4 bits (soit 16384 bits). Il est encore raisonnable d'intégrer S' dans un FPGA, mais

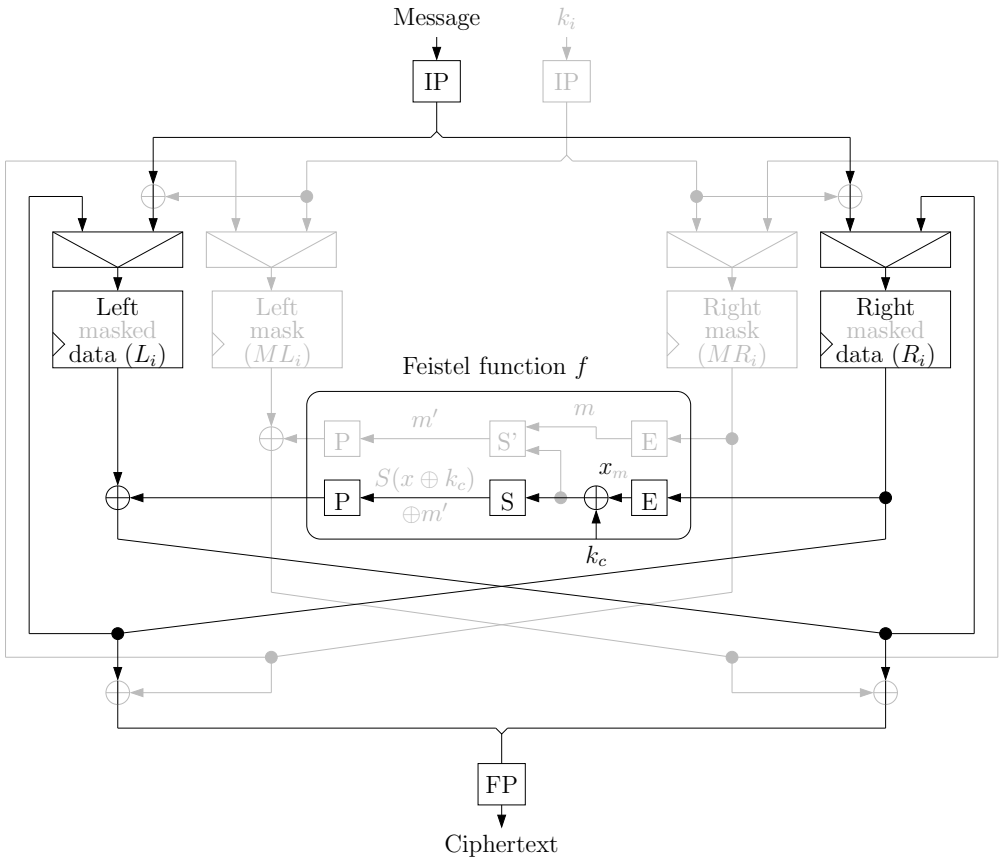


FIGURE 1.4 – Schéma du DES masqué avec deux chemins parallèles.

au prix d'une grande utilisation de blocs de mémoire (appelés BRAM). Par contre, ceci serait impossible pour un AES dont la sbox serait tabulée. Effectivement, $2^{16} \times 8$ bit = 524288 bit, ce qui est proche de la capacité totale de la mémoire d'un FPGA. Ainsi, il est plutôt préconisé de chercher une description de la sbox d'AES, appelée `SubBytes`, en sous-table. C'est ce qui est fait dans [381] avec une description de `SubBytes` non pas dans $\mathbb{F}_{2^8}$ mais dans $(\mathbb{F}_{2^4})^2$. Il est encore possible de descendre plus bas dans des extensions de corps [392], mais au prix d'une grande perte de débit, car chaque passage d'une extension à une autre demande de la logique combinatoire.

Il est également possible d'ajouter plus de masques, pour accroître la sécurité. Néanmoins, une table non protégée de taille n-bit d'adresse et m bits de sortie demandera $2^{(d+1)\times n} \times m$ bit de mémoire après protection avec d masques. Ainsi, pour les applications embarquées, où l'on ne souhaite pas avoir de perte de débit trop importante (plus les tables sont grosses, plus elles sont lentes), il faut se résoudre à n'utiliser qu'un seul masque. C'est ce que l'on appelle une CM de masquage du premier ordre.

1.3.2.1.2 Optimisation avec transformation du masque Dans le cas de figure où l'on ne peut pas augmenter le nombre de masques, on peut toutefois essayer d'accroître le niveau de sécurité. Supposons que les variables fuient au travers d'une fonction non-injective, comme le poids de Hamming. Alors, nous avons constaté qu'un attaquant pouvait fortement atténuer l'effet du masque m en considérant la somme ou la différence des deux fuites $\mathsf{HW}(x \oplus m)$ et $\mathsf{HW}(m)$. Notons X et M les variables aléatoires prenant les valeurs x et m. Typiquement, la variable $\mathsf{HW}(X \oplus M, M) = \mathsf{HW}(X \oplus M) + \mathsf{HW}(M)$ est déterministe quand $X = 255 =$ `0xff` (*i.e.* tous les bits de X sont à un). Effectivement, ceci est dû à la propriété $\mathsf{HW}(\neg m) = n - \mathsf{HW}(m)$, valable pour tout vecteur $m \in \mathbb{F}_2^n$.

Ainsi, il paraît opportun de mélanger mieux $X \oplus M$ et M pour éviter de tomber trop souvent dans cette identité remarquable qui gâche l'entropie de M. Nous avons ainsi proposé d'encoder le masque avec une bijection F, de telle sorte que les parties soient désormais $X \oplus M$ et $F(M)$. L'amélioration de la CM est représentée dans la Fig. 1.5 : S est la fonction cryptographique de ronde (typiquement une sbox) et R est la fonction de rafraîchissement de masque. Les parties combinatoires sont consignées en mémoire (*e.g.* RAM ou ROM), de manière à ne pas révéler autre chose que les entrées ou les sorties. Cela signifie concrètement que l'on suppose que les registres, contenant $X \oplus M$ et $F(M)$, sont les seules ressources à fuir de l'information.

Une représentation intuitive de l'effet de cette CM dite « *leakage squeezing* » (pour « compression des fuites ») est donnée dans la Fig. 1.6. On y voit que lorsqu'il existe un lien direct entre les données sensibles et la fuite, l'attaquant peut facilement inverser la fonction de fuite pour remonter aux données sensibles. C'est ce qu'il se passe lorsqu'il n'y a pas de protection (tiers supérieur de la Fig. 1.6). Le masquage vise à brouiller la fonction réciproque « fuite vers donnée sensibles », mais ne parvient pas nécessairement à homogénéiser la relation. Par exemple, dans le cas du masquage booléen du premier ordre (voir le tiers central de la Fig. 1.6) :

- la fuite $x = 0$ trahit sans équivoque la donnée sensible $y =$ `0x0`, ou bien
- la donnée sensible $y =$ `0xf` induira de façon déterministe la fuite $x = 4$, indépen-

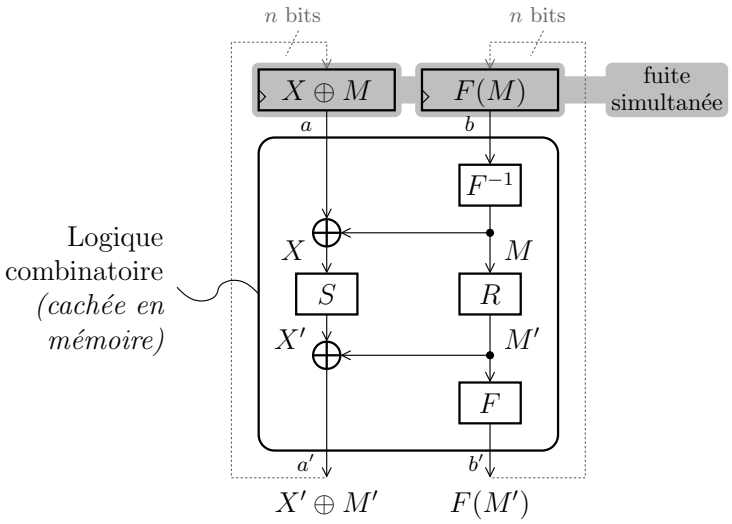


FIGURE 1.5 – Principe de l'amélioration du masquage du premier ordre par la technique du « *leakage squeezing* ».

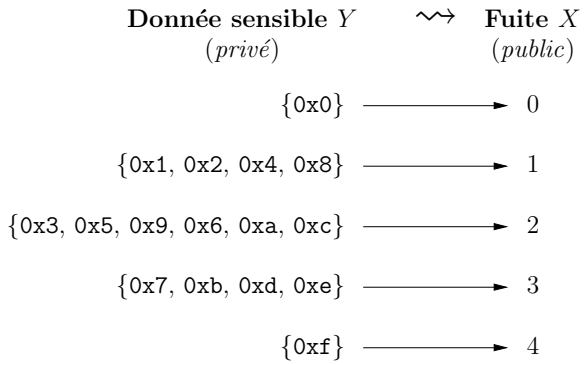
demment du masque M qui lui est appliqué.
La CM de « *leakage squeezing* » compresse les fuites en lissant les disparités (exploitables par l'attaquant) de la relation entre X et Y. La situation idéale est représentée dans le tiers inférieur de la Fig. 1.6. Néanmoins, cette figure ne transcrit de façon imagée qu'une intuition. Effectivement, en appliquant la CM de « *leakage squeezing* », les distributions $X \mid Y = y$ ne sont plus dégénérées pour les y de même poids de Hamming.

Notre étude [276] caractérise les meilleures fonctions F en terme de résistance aux attaques par analyse de corrélation (« *Correlation Power Analysis* », ou CPA [60]) d'ordre élevé. Comme expliqué dans [277], il s'agit de prendre F telle que $I||F : (x, y) \mapsto (x, F(y))$ soit un code de distance (directe) minimale maximale, et ait deux ensembles d'information complémentaires [67] (car F doit être bijective).

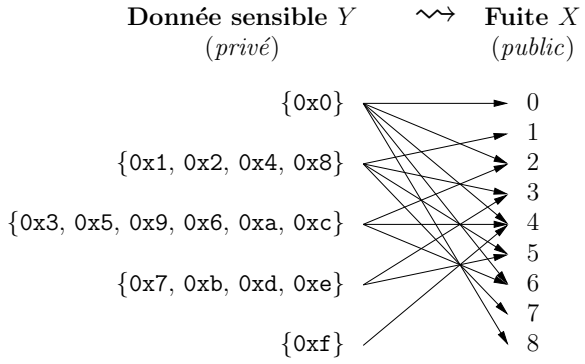
1.3.2.1.3 Optimisation avec annulation statistique des fuites Nous avons également recherché un schéma de masquage qui résisterait aux attaques CPA de tous les ordres. Une idée pour y parvenir est de constater que plus les deux parties fuient différemment, moins leur combinaison est efficace. Ainsi, un objectif pour le défenseur est de faire fuire au maximum une partie (donc de l'avoir d'entropie maximale), tandis que l'autre partie a une fuite inhibée (par exemple en la forçant à être déterministe). Il se trouve que, sous réserve que la fuite soit en distance (pas nécessairement de Hamming), il existe un tel schéma de masquage parfait [282] [3]. La fonction de fuite est notée, de façon

3. Ce papier est tenu à jour à cet emplacement [283]. Cette version présente notamment quelques corrections dans la construction des fonctions F de la Sec. 4.1.

Référence. Sans protection, on a : $X = \mathsf{HW}(Y)$. Notations : $Y \in \mathbb{F}_2^n$, d'où $\mathsf{HW}(Y) \in \{0, \cdots, n\}$. Il y a un lien trivial entre la fuite observée et la variable sensible.



Masquage booléen du premier ordre (tel que présenté dans la Fig. 1.4). Le modèle de fuite devient : $X = \mathsf{HW}(Y \oplus M) + \mathsf{HW}(M) \in \{0, \cdots, 2n\}$. Il subsiste toujours un lien entre les fuites observables et la variable sensible. Typiquement, si $Y = 2^n - 1 =$ `0xf` $(n = 4)$, le masque M ne remplit pas sa fonction de camouflage.



Masquage booléen du premier ordre avec « *leakage squeezing* » . Le modèle de fuite est alors plus complexe. $X = \mathsf{HW}(Y \oplus M) + \mathsf{HW}(F(M))$. Le rôle de la bijection F est de distribuer plus équitablement la variable sensible dans les $2n + 1$ classes de fuites (bien d'une distribution complètement équilibrée soit impossible).

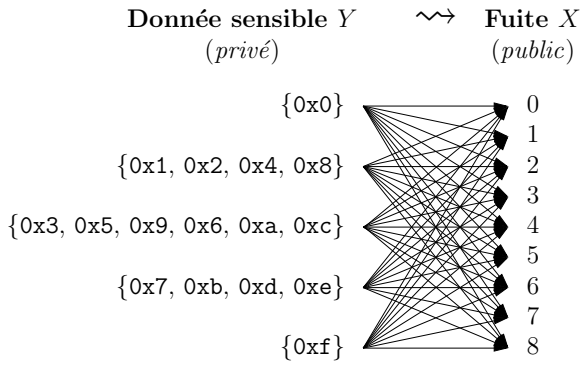


FIGURE 1.6 – Illustration de la CM « *leakage squeezing* », dans le cas $n = 4$.

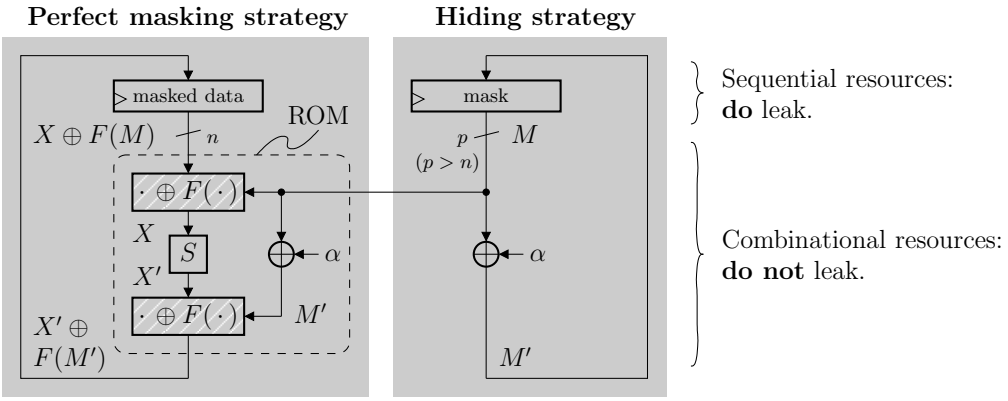


FIGURE 1.7 – Schéma de masquage avec annulation statistique des fuites.

abstraite,

$$\mathcal{A}(X, X') \;=\; A(X \oplus X') \;, \tag{1.2}$$

où X est la valeur initiale d'une variable et X' sa valeur finale. La CM marchera pour toute fonction de fuite $A : \mathbb{F}_2^n \to \mathbb{R}$. Une instanciation du schéma de masquage consiste à se limiter à seulement deux valeurs de masques, à savoir M et $M' = M \oplus \alpha$, pour une constante $\alpha \neq 0$ prédéterminée et publiquement divulguée. Le registre contenant le masque fuit alors :

- soit $\mathcal{A}(M, M \oplus \alpha) = A(M \oplus (M \oplus \alpha))$,
- soit $\mathcal{A}(M \oplus \alpha, M) = A((M \oplus \alpha) \oplus M)$,

i.e. une constante, à savoir $A(\alpha)$. Ensuite, le masque est injecté dans la donnée sensible depuis une fonction F, de sorte que le registre contenant la donnée masquée ait une fuite égale à $\mathcal{A}(X \oplus F(M), X' \oplus F(M')) = A(X \oplus X' \oplus F(M) \oplus F(M'))$. Le schéma de la contremesure est illustré dans la Fig. 1.7.

Si l'on note $\Delta X = X \oplus X'$ la (distance entre valeurs de la) variable sensible et Y la variable aléatoire $F(M) \oplus F(M \oplus \alpha)$, alors la fuite est en $A(\Delta X \oplus Y)$. Elle n'apporte aucune information sur ΔX si et seulement si $Y = F(M) \oplus F(M \oplus \alpha)$ est équilibrée (quand M suit une loi uniforme, *i.e.* $M \sim \mathcal{U}(\mathbb{F}_2^p)$). Or, cela est possible, par exemple si F est choisie telle que sa dérivée en α soit équilibrée. De telles fonctions Booléennes existent, mais il faut considérer que F envoie les éléments de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^p$, avec $p > n$. Effectivement, Y prend les mêmes valeurs pour m et $m \oplus \alpha$, $\forall m \in \mathbb{F}_2^n$. Des constructions pour $p = n + 1$, à base de fonctions à moitié nulle et de fonctions de type Maiorana-McFarland, sont explicitées dans [282]. La fonction $(X, M) \in \mathbb{F}_2^n \times \mathbb{F}_2^p \mapsto X \oplus F(M)$, représentée hachurée en blanc dans la Fig. 1.7, est aussi appelée "alpha" dans [282].

Comme annoté **en gras** au-dessus de la Fig. 1.7, on voit que le registre de masque est en fait protégé par une CM de dissimulation, tandis que celui accueillant la donnée masquée est quand à lui protégé par un masquage parfait. Ainsi, cette CM est une symbiose de deux paradigmes de protection, qui les associe tout en palliant leurs défauts :

– Le masquage d'ordre d est vulnérable aux attaques d'ordre $d+1$ si les $d+1$ parties fuient. Or cette CM veille à ne pas faire fuir une partie, en l'occurrence le registre de masque.
– Les logiques équilibrées, décrites dans la Sec. 1.5, procurent une bonne atténuation des fuites, à condition que l'implémentation ne présente pas de déséquilibre flagrant. Or précisément, la CM présentée dans la Fig. 1.7 dispose d'un chemin de mise à jour du masque qui est très simple, ce qui permet de rendre les différents bits bien indiscernables. Si la transformation $M' \leftarrow M$ était plus compliquée qu'un simple XOR, alors la partie droite de la Fig. 1.7 pourrait elle aussi être tabulée pour mieux équilibrer les bits entre eux (*cf.* notre préconisation de [42]).

1.3.2.2 Unique chemin de données avec tables en mémoire

Sans hypothèse aucune sur le modèle de fuite, il est possible d'avoir une fuite nulle (au sens de la théorie de l'information) si l'on considère que le calcul est effectué en mode homomorphique, *i.e.* sur la variable masquée uniquement. Comme expliqué précédemment, cela est possible avec du masque booléen du moment que l'on ne traverse pas de fonction non-linéaire. Quant aux sboxes, elles sont alors considérées comme des opérations où, tout à la fois :
– on démasque la variable,
– on applique la fonction non-linéaire et
– on remasque avec un masque frais.
Il est évidemment nécessaire que ces trois opérations ne soient pas réalisées séquentiellement, mais bien concomitamment, *i.e.* qu'elles soient fusionnées, de sorte que la variable sensible n'apparaisse pas dans une ressource matérielle. Pour plus d'efficacité, les sboxes masquées de part et d'autre sont précalculées, et stockées ainsi en mémoire. Elles viennent occuper une place de 2^8 mots de 8 bits en mémoire (quand les mots étudiés sont des octets).

Une version, illustrée sur AES, où uniquement 16 sboxes sont implantées en mémoire est décrite théoriquement dans [332] (*cf.* partie G à la page 205) et pratiquement dans [178, 334]. Le choix des sboxes s'effectue donc selon une variable aléatoire de 4 bit, qui détermine un décalage circulaire secret Γ de $[\![1, 16]\!]$. Un schéma simplifié du chemin de données d'un AES implémentant cette CM est donné à la Fig. 1.8. Les itérations de l'AES sont indiquées par l'entier i, qui démarre à zéro au début du calcul, et s'incrémente au fil des tours. Les 16 masques utilisés sont notés M_0, M_1, $\cdots$, M_f, où les indices sont imprimés en hexadécimal. Dans la figure, les indices sont à comprendre modulo 16. De plus, par souci de simplicité, uniquement le calcul des tables de substitution (`SubBytes`) de l'AES est représenté. On voit qu'un octet de l'état (*i.e.* l'un de X_0, X_1, $\cdots$, X_f) est masqué successivement avec des masques différents, d'indices égaux à Γ, $\Gamma + 1$, *etc.*

En réalité, même si l'on qualifie cette CM de masquage à chemin unique, il y a bien entendu en réalité une manipulation du masque à deux dates : une première fois pour déterminer quelle sbox est recherchée en mémoire, et une seconde fois lors de l'appel effectif de la sbox. La sécurité repose donc sur une bonne discrétion lors du choix de la sbox. Cette CM pourrait souffrir de ce type de défaut si elle n'était pas implémentée

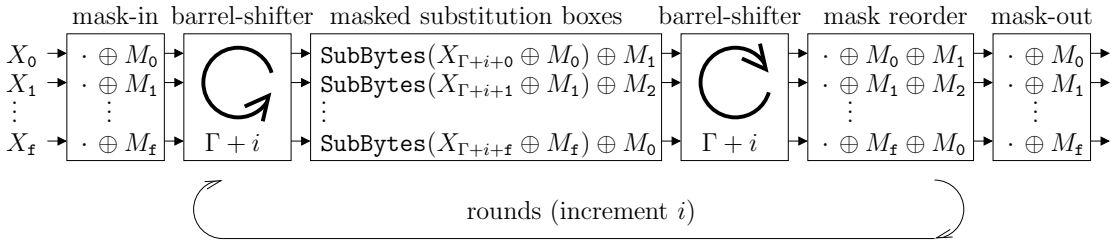


FIGURE 1.8 – Schéma de principe de la CM à chemin unique, avec « tables tournantes ».

sur un FPGA. Effectivement, si les sboxes étaient adressées individuellement, la variable aléatoire Γ pourrait fuir. Or en matériel, les 16 sboxes sont adressées en parallèle, et ce indépendemment de cette variable aléatoire, ce qui contribue donc à la dissimuler. Bref, on se rend compte que la sécurité s'appuie ici sur une combinaison de masquage (pour une part) et de dissimulation (pour l'autre). Ainsi, on retombe sur un principe de protection qui partage le même esprit que celui de la Sec. 1.3.2.1.3.

1.3.3 Conclusions

Le masquage s'implémente efficacement en matériel, et surtout en FPGA où il y a énormément de ressources disponibles. De plus, le débit n'est quasiment pas altéré. Les masques sont des variables aléatoires qui peuvent être produites par des générateurs d'aléa vrai (ou TRNG, pour « *True Random Numbers Generator* ») ou plus simplement par des registres à décalage à rebouclage linéaire, initialisés par une graine aléatoire.

Réalisé en matériel, il y a plusieurs façons d'améliorer la CM de masquage. Le tableau 1.1 résume les trois stratégies que nous inventées. Pour chacune d'entre elles, les hypothèses sur le modèle de fuite sont listées. Pour une efficacité optimale, les CM nécessiterons, selon le cas, un modèle en distance (comme dans l'Eqn. (1.2)), en poids de Hamming (HW) ou en distance de Hamming (HD). Essentiellement, si le modèle est tel qu'indiqué, avec un unique masque, la CM permet d'assurer le même niveau de protection qu'une CM à plus de masques. Le nombre équivalent de masques est évalué comme l'ordre maximal des attaques qui échouent contre les CM. Dans le cas d'un modèle qui dévie de celui qui est requis, l'ordre équivalent diminue.

1.4 Masquage : évaluation

Nous avons étudié à la Sec. 1.3 des schémas de protection où toutes les parties étaient consommées (*i.e.* traitées) simultanément. Ainsi, la fuite a lieu à la même date. Nous étudions par conséquent des attaques mono-variées. Il est à noter que l'état de l'art du masquage, surtout dans son implémentation logicielle, s'attaque préférentiellement avec des analyses multivariées. Ce type d'attaque pourrait également être appliqué à des montages d'acquisition multi-sonde, comme proposé par L. Sauvage *et al.* [396]. Néanmoins,

TABLE 1.1 – Comparaison des caractéristiques des différentes améliorations au masquage du premier ordre en matériel.

CM / **Sécurité**	**Bijection F, *aka* "*leakage squeezing*"** *cf.* §1.3.2.1.2	**Fonction *alpha*, *aka* "*Leak-free*"** *cf.* §1.3.2.1.3	**Masque peu entropique** *cf.* §1.3.2.2
Modèle	HW ou HD	Distance	HW ou HD
Ordre ($n = 4$ bit)	**3**	**4** (*i.e.* l'ordre de valeur maximale)	**2** (avec 8 valeurs de masque)
Ordre ($n = 8$ bit)	**5**	**8** (*i.e.* l'ordre de valeur maximale)	**2** (avec 12 valeurs de masque)

nous laissons volontairement ce type d'attaque de côté dans notre analyse. Nous mentionnons simplement que, du point de vue de la sécurité, la comparaison d'une implémentation à consommation parallèle et séquentielle des parties dépend du niveau de bruit N (supposé ici stationnaire pour simplifier le raisonnement). Effectivement, si l'on note $L_0 = \mathsf{HW}(Y \oplus M)$ et $L_1 = \mathsf{HW}(M)$ les fuites sans bruit des deux parties, alors, l'attaquant dispose respectivement :

- du scalaire $L_0 + L_1 + N \in \mathbb{R}$ pour une implémentation parallèle, et
- du couple $(L_0 + N, L_1 + N) \in \mathbb{R}^2$ pour une implémentation séquentielle.

Si dans le second cas l'attaquant peut réaliser des combinaisons, il est clair que la « somme » des deux fuites présente un rapport signal à bruit défavorable. Peut-être qu'une combinaison plus heureuse, comme le « produit centré » [365], peut améliorer le rapport signal à bruit, mais cela ne sera possible que si le bruit est faible. Une étude plus détaillée du compromis parallélisme / sécurité devrait donc être conduite pour trancher.

Cette section évalue le schéma de masquage décrit dans la Fig. 1.4. Notons X la fuite et Y la variable sensible. Lorsque l'hypothèse sur la variable sensible est correcte (*i.e.* l'attaquant a deviné la bonne clé), alors il existe un lien entre X et Y. Dans le cas où le circuit n'est pas protégé, nous prenons l'exemple d'une fuite en poids de Hamming. Dans le cas de la protection par masquage, nous gardons le même modèle, mais avec cette fois-ci une fuite concomitante du registre de la donnée masquée et du masque : $X = \mathsf{HW}(Y \oplus M, M) = \mathsf{HW}(Y \oplus M) + \mathsf{HW}(M)$. Par rapport à la Fig. 1.4, $Y \oplus M$ est R_i et M est MR_i, où $i \in [\![1, 16]\!]$ est l'indice de ronde. Lorsque l'attaquant se trompe de clé, le partitionnement fait intervenir une variable Y' qui est essentiellement indépendante de Y. La réalité est plus complexe, comme montré dans notre étude [194] (qui caractérise l'existence de pics fantômes, *i.e.* de corrélations non nulles même quand la clé est connue), mais ce niveau d'analyse simplifié est déjà intéressant. Si Y' dépend des variables cryptographiques, elle est uniformément distribuée. Ainsi, $Y' \sim \mathcal{U}(\mathbb{F}_2^n) \implies X = \mathsf{HW}(Y') \sim \mathcal{B}(n, 1/2)$. Effectivement, chacun des n bits est une variable aléatoire équi-répartie (*i.e.* obéissant à une loi de Bernoulli de paramètre $p = 1/2$), et leur somme suit donc une loi binomiale.

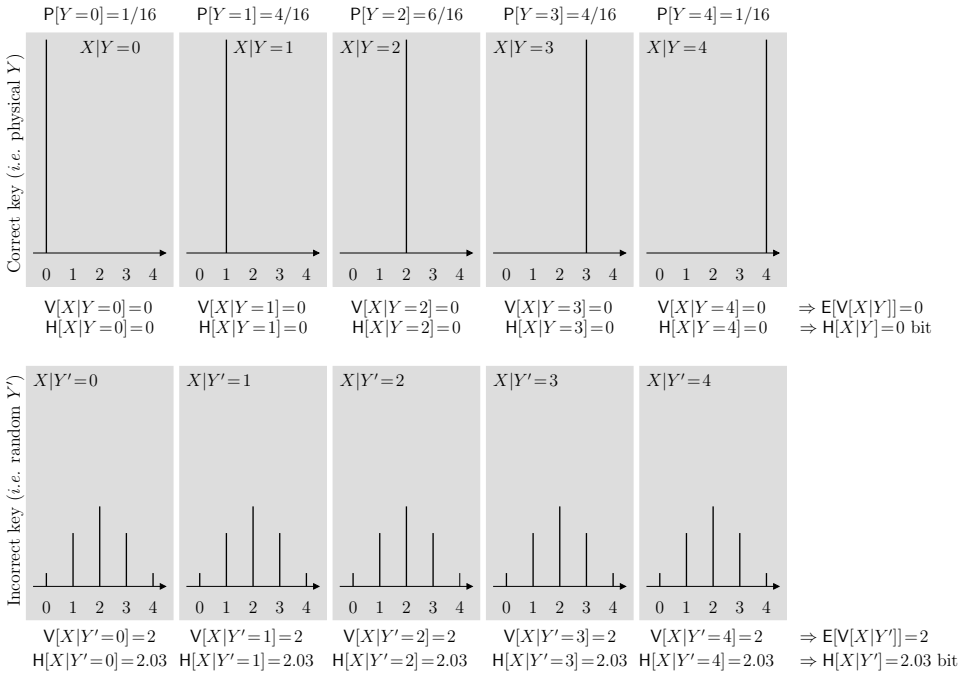


FIGURE 1.9 – PMF sans bruit en l'absence de contre-mesure.

Ce qui est remarquable, c'est que $X = \mathsf{HW}(Y')$ ne dépend pas de la vraie variable aléatoire Y. De même, dans le cas du masquage, la situation est quasiment comparable : $X = \mathsf{HW}(Y' \oplus M, M)$, avec $M \sim \mathcal{U}(\mathbb{F}_2^n)$. Ainsi $X \sim \mathcal{B}(2n, 1/2)$.

Les quatre cas étudiés sont les suivants :

1. Sans CM, bonne clé : ..... $X = \mathsf{HW}(Y)$.
2. Sans CM, mauvaise clé : ..... $X = \mathsf{HW}(Y')$.
3. Avec CM, bonne clé : ..... $X = \mathsf{HW}(Y \oplus M) + \mathsf{HW}(M)$.
4. Avec CM, mauvaise clé : ..... $X = \mathsf{HW}(Y' \oplus M) + \mathsf{HW}(M)$.

Les distributions sans bruit, aussi appelées PMF (*Probability Mass Functions*) sont données dans la Fig. 1.9 sans CM et dans la Fig. 1.10 avec CM. Les illustrations sont données pour $n = 4$, car cela permet de visualiser facilement les 5 classes. Les distributions conditionnelles de $X|Y$ (bonne clé) et $X|Y'$ (mauvaise hypothèse de clé) sont donc discrètes, car X ne prend que 5 valeurs, à savoir 0, 1, 2, 3, 4 ou 5.

1.4.1 Analyse de variance

Nous rappelons la loi de variance totale, énoncée dans la proposition 2 à la page 44 :

$$\underbrace{\mathsf{V}[X]}_{\text{Variance totale}} = \underbrace{\mathsf{E}[\mathsf{V}[X \mid Y]]}_{\text{Variance intra-classes}} + \underbrace{\mathsf{V}[\mathsf{E}[X \mid Y]]}_{\text{Variance inter-classes}} . \tag{1.3}$$

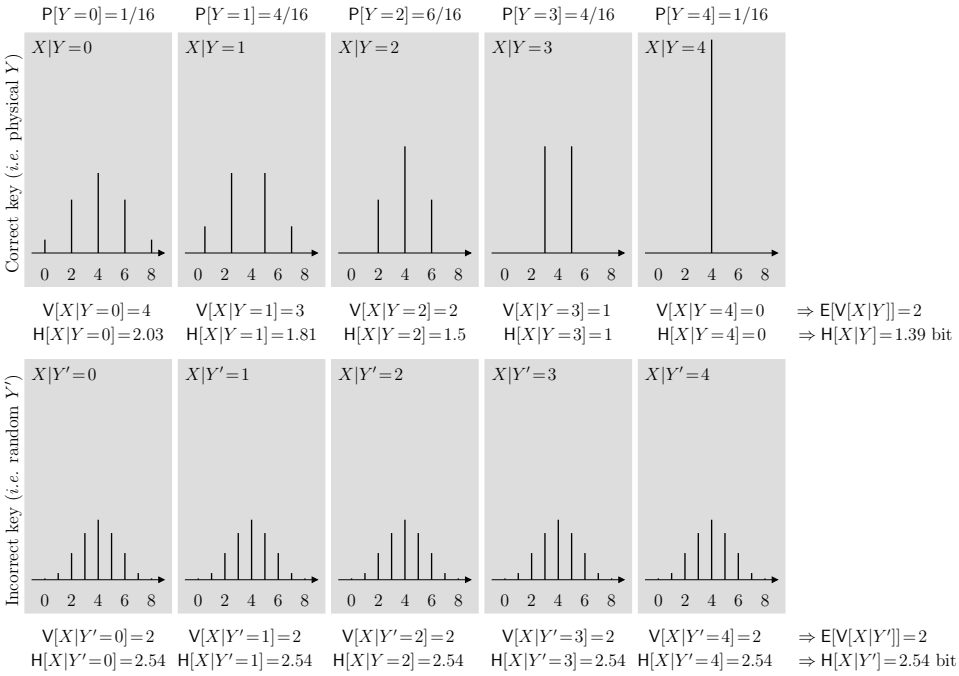


FIGURE 1.10 – PMF sans bruit sur la contre-mesure de la Fig. 1.4.

1.4.1.1 Sans bruit

On voit dans la Fig. 1.9 que, sans CM, la variance inter-classes augmente (respectivement : la variance intra-classes diminue) quand l'attaquant fait la bonne hypothèse de clé. À l'inverse, on voit dans la Fig. 1.10 qu'avec la CM, la variance inter-classes est nulle, que l'attaquant fasse ou non la bonne hypothèse de clé.

Dans [433] est introduit un test de variance (aussi présenté dans la Section III.C. de [142]). Il vise à maximiser le quotient $\mathsf{V}[X]/\mathsf{E}[\mathsf{V}[X \mid Y]]$. Cet objectif est en fait équivalent à minimiser la variance intra-classes $\mathsf{E}[\mathsf{V}[X \mid Y]]$. Effectivement, $\mathsf{V}[X]$ n'est pas sensible, étant donné que la variance totale ne dépend pas d'une hypothèse de clé en particulier.

1.4.1.2 Avec du bruit additif gaussien $\mathcal{N}(0, \sigma^2)$

La loi de l'Eqn. (1.3) s'étend au cas des signaux bruités, comme démontré dans l'Eqn. (1.17) de la Sec. 1.9.3.1. Ainsi, les conclusions restent semblables : la variance inter-classes ou intra-classes n'est un distingueur qu'en l'absence de contremesure.

1.4.2 Analyse en information mutuelle

On peut chercher une loi semblable au partitionnement donné dans la proposition 2, mais transposée au cadre de théorie de l'information. La décomposition s'écrit :

$$\underbrace{\mathsf{H}[X]}_{\text{Entropie}} = \underbrace{\mathsf{I}[X;Y]}_{\text{Information mutuelle}} - \underbrace{\mathsf{H}[X \mid Y]}_{\text{Entropie conditionnelle}} . \tag{1.4}$$

Cette décomposition est souvent représentée de façon imagée par le schéma de la Fig. 1.16.

Il existe effectivement un parallèle entre :

- **Variance totale** et **entropie** : le premier mesure la dispersion et le second l'incertitude ;
- **Variance inter-classes** et **information mutuelle** : le premier mesure la variation expliquée par la variable conditionnante et le seconde la quantité d'information apportée par la variable conditionnante ;
- **Variance intra-classes** et **entropie conditionnelle** : le premier mesure la dispersion résiduelle et le second la réduction d'entropie, dans les deux cas suite à la connaissance de la variable conditionnante.

1.4.2.1 Sans bruit

Sans bruit, il y a une différence d'information mutuelle entre les distributions pour la bonne clé et pour les mauvaises. Cette différence est plus faible, mais strictement non-nulle, quand la CM est appliquée. Cela positionne donc la MIA (pour « *Mutual Information Analysis* ») parmi les distingueurs efficaces (*i.e.* dont on peut prouver le principe de fonctionnent par la théorie).

1.4.2.2 Avec du bruit additif gaussien $\mathcal{N}(0, \sigma^2)$

Cette fois-ci, on considère que les mesures X sont bruitées. Elles suivent donc la loi décrite dans les PMF de la Fig. 1.10, à ceci près qu'elles sont de plus affectées d'un bruit additif $N \sim \mathcal{N}(0, \sigma^2)$. Leur somme devient donc une loi qui suit une distribution continue, ou PDF (*Probability Density Functions*), convolée des deux distributions. On parle de « mixture de gaussiennes ».

Dans les composants matériels, les calculs sont effectués en parallèle. Par exemple, dans un DES itératif, comme celui décrit dans [195], les registres d'état LR (64 bits) et de clé CD (56 bits) commutent simultanément. Ainsi, l'activité des 4 bits étudiés (sortie d'une sbox $6 \mapsto 4$) est à rapporter à l'activité décorrélée des $64 + 56 - 4$ autres bits. De plus, dans une implémentation pipelinée [435], le nombre de ces registres est démultiplié autant de fois que les boucles sont déroulées pour accroître le débit. Par conséquent, nous nous trouvons dans le cas où le bruit est notoirement plus important que le signal de fuite dû aux commutations de la variable sensible. Ceci signifie que $\sigma^2_{\text{tot}}, \sigma^2_y \ll \sigma^2$ (nous utilisons ici les raccourcis d'écriture suivants $\sigma_{\text{tot}} = \mathsf{V}[X]$ et $\sigma_y = \mathsf{V}[X \mid Y = y]$, qui seront aussi ré-introduits en annexe dans la Sec. 1.9.3.2). Ainsi, on peut développer l'expression de $\mathsf{I}[X + N; Y]$ (voir l'Eqn. (1.18) en annexe) en l'une des

variables $\epsilon = \sigma^2_{\text{tot}}/\sigma^2$ ou σ^2_y/σ^2. Ceci revient à peu près au même, car σ^2_{tot} est la moyenne des σ^2_y (si la variance inter-classe est nulle, ce qui est bel et bien l'objectif minimal attendu de tout schéma de masquage) et donc ces variances sont du même ordre de grandeur. L'approximation est clairement visible sur les PDF en présence de grand bruit, représentées dans la Fig. 1.11. Les partitionnements pour des Y de poids de Hamming égaux sont identiques. Ainsi, la Fig. 1.11 ne montre-t-elle que $n+1=5$ PDF. Pour des petits bruits, inférieurs à l'unité (*i.e.* uniquement l'activité d'un seul bit sensible), on reconnaît les PMF théoriques, montrées dans la Fig. 1.10. Quand le bruit est supérieur à l'unité, les distributions ne dévoilent plus, du moins visuellement, la quantification des classes. Quand le bruit est beaucoup plus grand que l'unité, les mixtures de gaussiennes tendent vers des gaussiennes[4], ce qui légitime nos approximations.

Il est intéressant de quantifier l'erreur commise en substituant dans les calculs les distributions $X \mid Y = y$ par $\mathcal{N}(\mathsf{E}[X \mid Y = y], \mathsf{V}[X \mid Y = y])$. Un outil approprié est la divergence de Kullback-Leibler (voir la définition en Sec. 1.9.1.2 à la page 44). Elle est tracée dans la Fig. 1.12, pour les 5 classes de poids de Hamming de Y. Nous savons que, quand $y = \texttt{0xff}$, le masquage n'enrichit pas la distribution. Ainsi, $X \mid Y = \texttt{0xff}$ est déjà une gaussienne : sa divergence de Kullback-Leibler par rapport à son approximation parfaite est donc nulle. Quant aux autres valeurs de $\mathsf{HW}(y)$, on observe deux zones :

1. Pour les faibles bruits ($\sigma < 1$), la divergence est d'autant plus grande que $\mathsf{HW}(y)$ est grand. Effectivement, comme déjà mentionné, il est connu qu'une loi binomiale tend vers une loi gaussienne quand le nombre de valeurs augmente (bien sûr, il faut, comme nous l'avons fait, exclure le cas singulier à une unique valeur).
2. Pour de grands bruits ($\sigma > 1$), alors la divergence est d'autant plus grande que $\mathsf{HW}(y)$ est petit. La tendance est donc inversée. L'explication est que plus il y a de composantes dans une mixture de gaussiennes, plus il est délicat de l'assimiler à une gaussienne [389].

Cependant, au-delà de ces différences relatives de divergences par classe $\mathsf{HW}(y) \in \{0, \cdots, 4\}$, on constate que quelque soit la classe, la divergence diminue environ exponentiellement quand the bruit croît.

L'expression de l'information mutuelle entre $X+N$ et Y est donnée dans l'Eqn. (1.18) de la Sec. 1.9.3.2. Au premier ordre, on utilise le développement limité $\ln(1+\epsilon) = \epsilon + \mathcal{O}(\epsilon)$, valide quand $\epsilon \longrightarrow 0$. Il donne l'expression suivante :

$$\mathsf{I}[X+N;Y] = -\frac{1}{2\ln 2}\left(\frac{\sum_{y\in\mathcal{Y}} \mathsf{P}[y]\cdot\sigma^2_y - \sigma^2_{\text{tot}}}{\sigma^2}\right) = \frac{1}{2\ln 2}\frac{\mathsf{V}[\mathsf{E}[X \mid Y]]}{\mathsf{V}[N]} + \mathcal{O}\left(\frac{\mathsf{V}[X]}{\mathsf{V}[N]}\right). \tag{1.5}$$

Ainsi, si la variance inter-classes est nulle, l'information mutuelle est nulle au premier ordre.

De même, on peut calculer une approximation de l'entropie conditionnelle. Comme d'après l'hypothèse gaussienne, $X + N \mid Y = y \sim \mathcal{N}(0, \mathsf{V}[X+N \mid Y = y])$, on a au

4. Cette observation peut aussi se voir comme l'approximation de la PMF par une gaussienne, étant donné l'importance du bruit N. Or il est bien connu que la somme de deux gaussiennes suit une loi gaussienne.

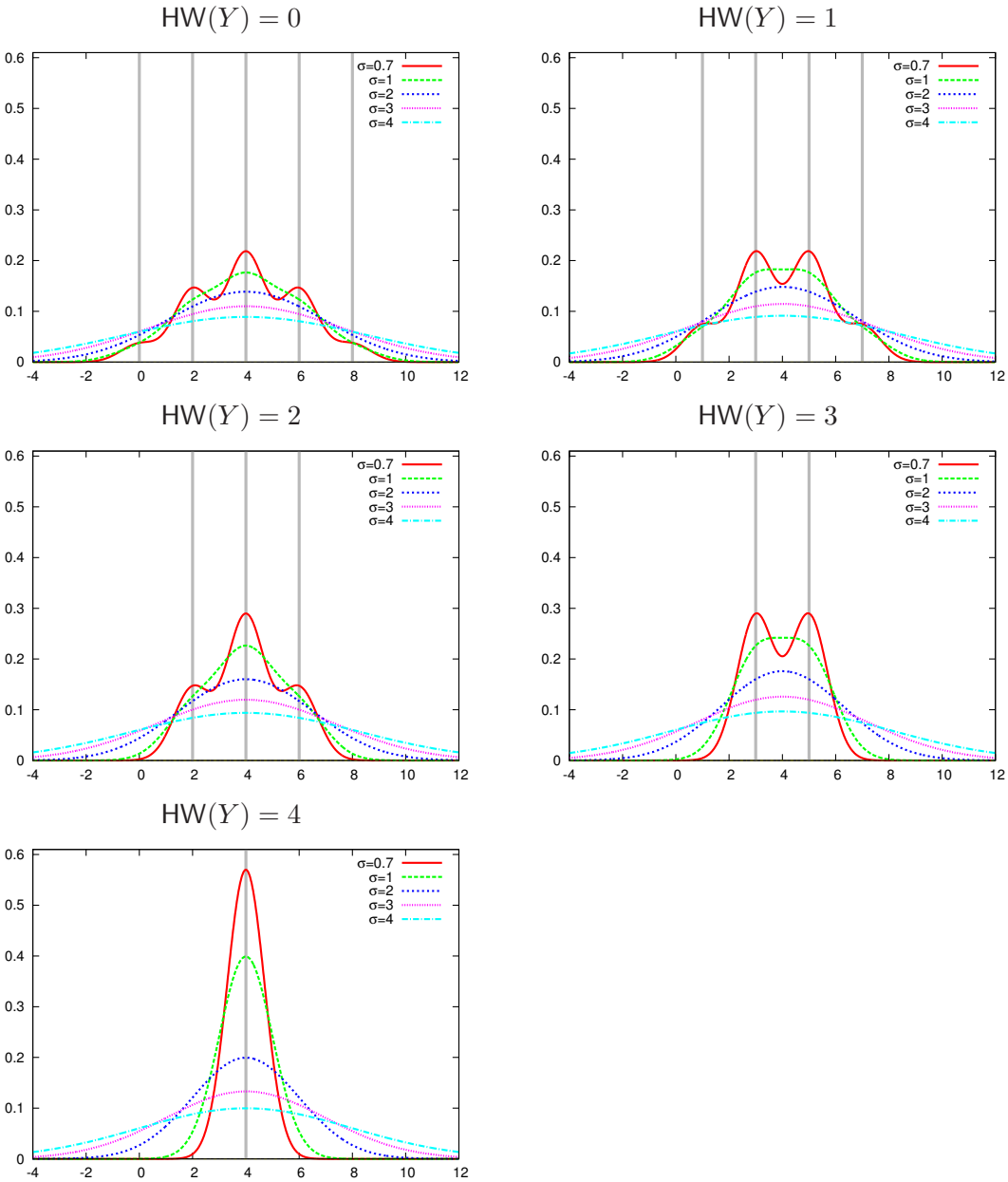


FIGURE 1.11 – PDF de la CM pour la fuite $X = \mathsf{HW}(Y \oplus M) + \mathsf{HW}(M)$ sur $n = 4$ bit. L'abscisse représente les valeurs prises par $X + N$, où $N \sim \mathcal{N}(0, \sigma^2)$ est un bruit additif. À titre de repère, des barres verticales grises indiquent les valeurs discrètes prises par X.

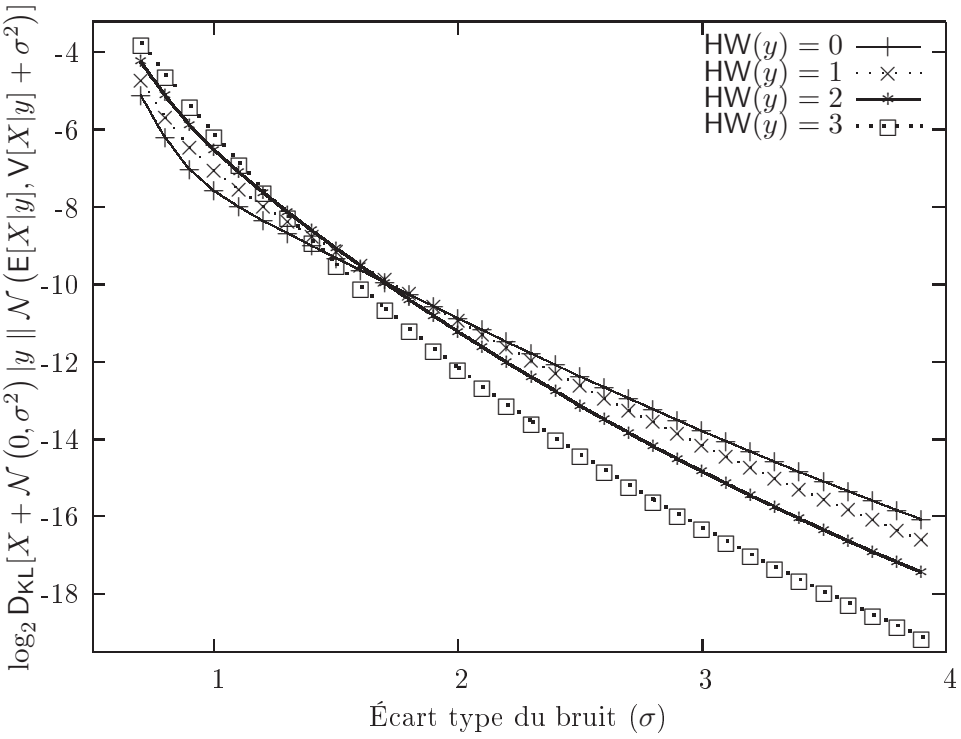


FIGURE 1.12 – Divergence de Kullback-Leibler entre les vraies PDF $X + N \mid Y = y$ et leur approximation gaussienne.

premier ordre [364] :

$$\begin{aligned}
\mathsf{H}[X+N \mid Y=y] &= \frac{1}{2}\log_2\left(2\pi e \mathsf{V}[X+N \mid Y=y]\right) \quad // \textit{ Cf.} \text{ Eqn. (1.11)} \\
&= \frac{1}{2}\log_2\left(2\pi e \left(\mathsf{V}[X \mid Y=y]+\mathsf{V}[N]\right)\right) \quad // \textit{ Cf.} \text{ Eqn. (1.9) \& (1.10)} \\
&= \frac{1}{2}\log_2\left(2\pi e \mathsf{V}[N]\right) + \frac{1}{2\ln 2}\frac{\mathsf{V}[X \mid Y=y]}{\mathsf{V}[N]} + \mathcal{O}\left(\frac{\mathsf{V}[X]}{\mathsf{V}[N]}\right).
\end{aligned}$$

Sur cette formule linéarisée, on peut effectuer une somme pondérée par les $\mathsf{P}[y], y \in \mathcal{Y}$. On obtient :

$$\mathsf{H}[X+N \mid Y] = \frac{1}{2}\log_2\left(2\pi e \mathsf{V}[N]\right) + \frac{1}{2\ln 2}\frac{\mathsf{E}[\mathsf{V}[X \mid Y]]}{\mathsf{V}[N]} + \mathcal{O}\left(\frac{\mathsf{V}[X]}{\mathsf{V}[N]}\right). \tag{1.6}$$

Ces deux expressions Eqn. (1.5) et (1.6) sont cohérentes entre elles : on a bien une conservation au niveau théorie de l'information (Eqn. (1.4)) grâce à la conservation au niveau de la variance (Eqn. (1.3)).

Elles montrent que les conclusions tirées en l'absence de bruit (Sec. 1.4.2.1) ne sont plus vraies avec du bruit. Effectivement, au premier ordre en $\mathsf{V}[X]/\mathsf{V}[N]$, avec CM, l'information mutuelle s'annule aussi bien pour la bonne clé que pour les mauvaises.

1.4.3 Évaluation

Le tableau 1.2 résume les résultats obtenus dans cette section 1.4 sur l'évaluation du masquage. On voit qu'une variance inter-classes non nulle permet de distinguer une bonne hypothèse de clé d'une mauvaise, en l'absence de contremesure. Cette spécificité a d'ailleurs été mise à profit dans notre attaque *First Principal Component Analysis* (FPCA [430]). Non seulement la PCA a été utilisée pour trouver les instants de fuite (comme décrit dans [13]), mais également pour maximiser la variance inter-classes due au lien entre X et Y. On constate également que la MIA fonctionne comme un distingueur dans ce cas précis pour les mêmes raisons. Comme évoqué dans le rapport [238], la *Linear Discriminent Analysis* (LDA) peut également jouer un rôle comparable à la PCA. Alors que la PCA maximise la variance inter-classes, la LDA se sert astucieusement du corollaire qui est que concomitamment, la variance intra-classes diminue. Ainsi, pour capturer les deux tendances, la LDA en réalise le ratio.

Enfin, quand la contremesure est activée, la variance inter-classes est annulée, ce qui conduit effectivement à l'échec de ces deux analyses.

L'idée de montrer la similarité entre les attaques mono-variées est aussi décrite dans l'article [291, 292] ; il détaille les conditions (notamment en présence de bruit asymptotiquement grand) pour lesquelles on peut considérer que tous les distingueurs sont équivalents, dans le sens que l'écart du nombre de traces nécessaires pour retrouver le secret diminue. Par ailleurs, l'idée de montrer que certaines attaques sont des approximations d'autres a été présentée dans l'article [256] ; il montre que la CPA s'obtient naturellement comme le développement à l'ordre un de la MIA. Il prouve aussi que les attaques

TABLE 1.2 – Distingueurs par moments statistiques ou par théorie de l'information, dans l'approximation gaussienne, et au premier ordre, *i.e.* $\mathcal{O}\left(\mathsf{V}[X]/\mathsf{V}[N]\right)$.

Contexte	Variance inter-classes $\mathsf{V}[\mathsf{E}[X+N \mid Y]]$	Variance intra-classes $\mathsf{E}[\mathsf{V}[X+N \mid Y]]$	Variance totale $\mathsf{V}[X+N]$
Générique	$\mathsf{V}[\mathsf{E}[X \mid Y]]$	$\mathsf{E}[\mathsf{V}[X \mid Y]] + \mathsf{V}[N]$	$\mathsf{V}[X] + \mathsf{V}[N]$
Sans CM, bonne clé	$\mathsf{V}[X]$	$\mathsf{V}[N]$	$\mathsf{V}[X] + \mathsf{V}[N]$
Sans CM, mauvaise clé	0	$\mathsf{V}[X] + \mathsf{V}[N]$	$\mathsf{V}[X] + \mathsf{V}[N]$
Avec CM, bonne clé	0	$\mathsf{V}[X] + \mathsf{V}[N]$	$\mathsf{V}[X] + \mathsf{V}[N]$
Avec CM, mauvaise clé	0	$\mathsf{V}[X] + \mathsf{V}[N]$	$\mathsf{V}[X] + \mathsf{V}[N]$

Contexte	Information mutuelle $\mathsf{I}[X+N;Y]$	Entropie conditionnelle $\mathsf{H}[X+N \mid Y]$	Entropie totale $\mathsf{H}[X+N]$
Générique	$\frac{1}{2\ln 2}\frac{\mathsf{V}[\mathsf{E}[X\|Y]]}{\mathsf{V}[N]}$	$\frac{1}{2}\log_2\left(2\pi e\mathsf{V}[N]\right) + \frac{1}{2\ln 2}\frac{\mathsf{E}[\mathsf{V}[X\|Y]]}{\mathsf{V}[N]}$	$\frac{1}{2}\log_2\left(2\pi e(\mathsf{V}[N]+\mathsf{V}[X])\right)$
Sans CM, bonne clé	$\frac{1}{2\ln 2}\frac{\mathsf{V}[X]}{\mathsf{V}[N]}$	$\frac{1}{2}\log_2\left(2\pi e\mathsf{V}[N]\right)$	$\frac{1}{2}\log_2\left(2\pi e(\mathsf{V}[N]+\mathsf{V}[X])\right)$
Sans CM, mauvaise clé	0	$\frac{1}{2}\log_2\left(2\pi e\mathsf{V}[N]\right) + \frac{1}{2\ln 2}\frac{\mathsf{V}[X]}{\mathsf{V}[N]}$	$\frac{1}{2}\log_2\left(2\pi e(\mathsf{V}[N]+\mathsf{V}[X])\right)$
Avec CM, bonne clé	0	$\frac{1}{2}\log_2\left(2\pi e\mathsf{V}[N]\right) + \frac{1}{2\ln 2}\frac{\mathsf{V}[X]}{\mathsf{V}[N]}$	$\frac{1}{2}\log_2\left(2\pi e(\mathsf{V}[N]+\mathsf{V}[X])\right)$
Avec CM, mauvaise clé	0	$\frac{1}{2}\log_2\left(2\pi e\mathsf{V}[N]\right) + \frac{1}{2\ln 2}\frac{\mathsf{V}[X]}{\mathsf{V}[N]}$	$\frac{1}{2}\log_2\left(2\pi e(\mathsf{V}[N]+\mathsf{V}[X])\right)$

peuvent être améliorées si, à l'inverse, on garde plus de termes dans les développements limités. Dans ce cas, les attaques en théorie de l'information deviennent pertinentes, car les distingueurs contiennent des termes à tous les ordres.

1.4.4 Conclusion

L'objet de cette évaluation du masquage a été de montrer que l'efficacité du masquage permet de résister à des attaques qui analysent la dispersion (en l'occurrence la variance) des traces X. Par la même occasion, il est démontré que des distingueurs plus génériques, comme l'information mutuelle, permettent tout de même de mettre en défaut la contremesure. C'est tout à fait notable lorsqu'il n'y a pas de bruit. Mais quand il existe un bruit additif gaussien, l'information mutuelle devient en première approximation proportionnelle à la variance inter-classes, qui est justement annulée par la contremesure.

C'est pour cette raison que de nouveaux distingueurs ont été proposés. À titre d'exemple, plutôt que de comparer le résultat d'un distingueur appliqué d'une part aux observations non partitionnées et d'autre part aux observations partitionnées, nous préconisons d'étudier la dispersion dans les classes. En analyse de variance, nous avons mis en avant la *Variance Power Analysis* (VPA [274], aussi suggérée indépendemment dans [433]), qui étudie la variance de la variance intra-classes. Du point de vue de la théorie de l'information, nous avons également appliqué la même idée, qui s'appuie sur une information mutuelle pondérée, ce qui permet de mettre en évidence des différences d'entropie entre les classes. L'attaque en question s'appelle l'*Entropy-based Power Attack* (EPA [281]). Elle est paramétrique, et permet notamment d'extraire une information différentielle même quand la variance inter-classes est nulle. Finalement, nous mentionnons qu'une source d'inspiration a été le papier présentant la *Differential Cluster Analysis* (DCA [22]). Cette publication présente le compromis inter/intra sur lequel l'attaquant peut jouer pour optimiser son attaque. Dans notre cas, l'idée est d'utiliser au mieux les variations intra-classes quand les variations inter-classes ont été annulées par la contremesure.

1.5 Dissimulation : principe

Dans cette section sont étudiées les logiques à activité constante statiquement. Le principe est l'équilibrage de l'activité globale. Dans la plupart des cas, on pourra même redescendre cette contrainte au niveau local (*i.e.* chaque instance a un comportement équilibré, indépendamment des données).

1.5.1 Niveau logique

Équilibrer l'activité au niveau logique consiste à changer l'encodage des données, qui doit être tel que le nombre de transitions soit indépendant des données. Ce problème est très général et bien connu par la communauté des codeurs. Par exemple, le codage de Gray permet de répondre à la contrainte de l'activité constante.

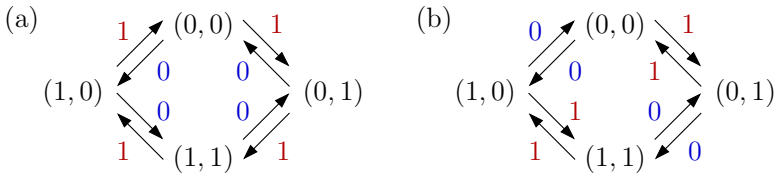


FIGURE 1.13 – Deux types d'encodage à activité constante.

Un code de Gray satisfait la propriété que deux mots de codes voisins ne diffèrent que d'un seul bit. Formellement, appelons G le code $G : \mathbb{F}_2^n \to \mathbb{F}_2^n$. Alors $\forall x \in \mathbb{F}_2^n, \mathsf{HW}(G(x) \oplus G(x \pm 1)) = 1$. Une telle fonction G est illustrée pour l'encodage d'un bit dans la Fig. 1.13(a). Un mot de $x \in \mathbb{F}_2^2$ est l'état, qui permet d'en déduire le codage $G(x)$ représenté comme un couple $(0/1, 0/1)$ dans la Fig. 1.13(a) ; le calcul d'une valeur 1 (resp. 0) est représenté par l'incrément (resp. le décrément) de l'état x. Ainsi, de façon exhaustive :

- les transitions parmi $\{(0,0) \to (0,1), (0,1) \to (1,1), (1,1) \to (1,0), (1,0) \to (0,0)\}$ représentent un 1, et
- celles qui ont lieu dans le sens contraire, *i.e.* dans l'ensemble $\{(0,0) \to (1,0), (1,0) \to (1,1), (1,1) \to (0,1), (0,1) \to (0,0)\}$, représentent un 0.

L'inconvénient de cette représentation est qu'une activité sur l'un des fils peut signifier un zéro ou un un selon l'état courant.

Ainsi, un encodage où chaque bit représente une valeur serait préférable. Essentiellement, cette représentation pourrait se mettre sous la forme suivante : (x_f, x_t), où les changements de x_f (resp. x_t) indiquent une prise de valeur 0 (resp. 1). Il existe, et est illustré dans la Fig. 1.13(b). On le qualifie quelquefois de « deux-phases ». Mais il présente encore un inconvénient, à savoir d'être à état. D'une part, au démarrage, il n'y a pas de valeur prédéterminée. D'autre part, dans un état donné, on ne connaît pas la valeur représentée si l'on n'a pas suivi tout le parcours de la variable.

C'est pour cette raison que l'on préfère finalement un autre encodage, dit « quatre-phases », illustré dans la Fig. 1.14. Il fait intervenir une phase de précharge en sus de la phase d'évaluation. La valeur prise pendant la précharge est $(0,0)$ (ou $(1,1)$, au choix [420]) et est notée NULL ; quand on utilise deux valeurs de précharge, on peut préciser $(0,0) =$ NULL0 ou $(1,1) =$ NULL1. Les deux valeurs valides sont $(x_f, x_t) = (1,0)$, ou VALID0, et $(x_f, x_t) = (0,1)$, ou VALID1. Cette fois-ci, l'encodage cumule les deux propriétés souhaitées :

1. **Séparabilité** : il y a un fil qui encode la valeur fausse, à savoir x_f (l'indice f signifie *false*), et un autre la valeur vraie, à savoir x_t (l'indice t signifie *true*) ;
2. **Sans état** : on peut déduire la valeur représentée par la lecture de (x_f, x_t), sans avoir à se souvenir de sa (ou ses) valeur(s) passée(s).

Les domaines où ce type d'encodage se rencontre sont variés. En plus de la sécurité, où l'on recherche l'indépendance de l'activité et des données (mais l'activité dépend du temps), on peut citer également le calcul asynchrone [328]. Il y a consensus dans ces

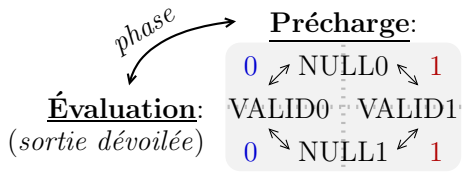


FIGURE 1.14 – Encodage séparable à précharge, engendrant une activité constante.

communautés pour l'utilisation de l'encodage « quatre-phase » présenté dans la Fig. 1.14. Maintenant, il s'agit de s'intéresser à la façon de calculer avec cet encodage, que l'on appelle traditionnellement « logique ». De manière générique, on parlera de DPL (*Dual-rail with Precharge Logic*).

Beaucoup de DPL ont été étudiées. Nous en avons réalisé un aperçu dans [97]. Plutôt que de présenter des exemples de logiques publiées, nous nous attachons à en définir les caractéristiques qui les différencient.

- Tout d'abord, on peut noter qu'il est possible de partir de paires différentielles toutes NULL et d'arriver à un état VALID (soit VALID0, soit VALID1), en imposant simplement aux portes logiques de respecter la condition de propager les NULL (resp. les VALID) lorsque toutes les entrées sont NULL (resp. VALID). Cette contrainte reflète le besoin d'une mise à NULL par vague (toutes les portes évaluant à NULL), et de la fonctionnalité du circuit (il calcule une value VALID au moins quand toutes les entrées sont VALID). Or, ce schéma peut provoquer des transitions temporaires non-fonctionnelles, appelées *glitches* ou *hazards*. Ces transitions surviennent à cause de différences de délais dans des portes en logique non-positive. Les *glitches* sont une source non-négligeable de fuite [372], car on estime qu'ils contribuent à environ 40 % de l'activité des circuits CMOS [270]. Des attaques se focalisant sur les *glitches* [295] et les incluant dans un modèle de fuite [268] ont déjà été démontrées.
- Les différences de délais peuvent également être la cause d'une propagation anticipée, soit en phase de précharge, soit en phase d'évaluation. On qualifie ce problème d'EPE (*Early Propagation Effect*). Il s'agit d'un effet qui a des origines « logiques », dans le sens que sans précaution particulière, il est possible d'anticiper la sortie d'une porte logique sans en connaître toutes les entrées[5]. Ainsi, il est possible que la date d'évaluation dépende des valeurs logiques. Cet effet a été caractérisé dans notre article [172] par des simulations logiques rétro-annotées (avec SDF [226]) en délais de propagation dans les portes et le réseau d'interconnexion.
- L'interconnexion des portes entre elles peut être déséquilibrée, car les outils de P&R ne sont pas nécessairement au courant que la *netlist* traitée est en DPL. Ce

5. Transposé au génie logiciel, c'est ce que l'on appelle une évaluation en court-circuit. Par exemple, en langage C, `a && b` s'évalue à faux si `a` est faux, sans même calculer `b`. On a le même cas de figure pour l'évaluation de `a || b` lorsque `a` est vrai. Il y a également des effets de bords qu'il faut considérer avec beaucoup d'attention.

biais déséquilibre une paire, et introduit donc une différence entre les fils portant la valeur sensible. Une simple modélisation de la dissipation liée au routage permet de quantifier ce phénomène. Si, en l'absence de protection, une ligne x est vue comme une capacité C_x commutant de 0 à V, l'attaquant doit distinguer entre une consommation de 0 ou de $\frac{1}{2}C_xV^2$. Avec la protection, l'attaquant doit distinguer entre les fuites $\frac{1}{2}C_{x_f}V^2$ et $\frac{1}{2}C_{x_v}V^2$. Ainsi, la différence vaut respectivement $\frac{1}{2}C_xV^2$ (sans protection) et $\frac{1}{2}\left(C_{x_f} - C_{x_v}\right)V^2$ (avec protection). La protection est efficace si $\left|C_{x_f} - C_{x_v}\right| < C_x$. Notons que cette condition est considérée en valeur absolue, car, *a priori*, il y a autant de chances que le déséquilibre soit du côté f que v.

Les mesures à prendre pour contrer les problèmes sont listées ci-après :

- Contre les *glitches*, on peut soit se limiter à des instances positives (qui sont monotones et donc dépourvues de *glitches* si les entrées sont également monotones, *cf.* notre analyse [204]), soit supprimer les différences de délais entre les signaux (*cf.* Isolated WDDL [300], par exemple).
- Contre l'EPE, il faut synchroniser les portes [331, 218] : la validité de leur sortie doit toujours être la conséquence d'une unanimité des validités en entrée. Quant au retour à la phase de précharge, il peut soit être toujours anticipé (car on connaît la valeur calculée, à savoir NULL), soit être toujours retardé.
- L'équilibrage des fils dans une paire peut être obtenu soit par un "masquage", soit par des contraintes physiques. Un effort sur les deux plans simultanément est bénéfique, car ces techniques peuvent se combiner constructivement. La Sec. 1.5.2 suivante détaille ce point.

1.5.2 Niveau physique

1.5.2.1 ASIC

Les outils de conception des ASIC permettent une très grande personnalisation. Le concepteur peut donc poser des contraintes pour conserver l'équilibrage logique au niveau physique. Aujourd'hui, aucun outil n'intègre la fonctionnalité de placement et de routage différentiel. Mais il est possible tout de même d'utiliser astucieusement les outils pour arriver à ses fins. Deux approches ont été proposées :

1. Duper l'outil : on l'instruit de placer et router un circuit "simple-rail", que l'on dédouble ensuite manuellement [457]. La tactique repose donc sur un changement des règles de dessin (les interconnexions sont plus épaisses).

2. Guider l'outil : on lui demande de placer et router la moitié du circuit, en ayant au préalable réservé la moitié des ressources pour une duplication postérieure [191]. Les contraintes sont de deux natures : (*i*) bloquages pour le placement et (*ii*) obstructions pour le routage.

Dans les deux cas, les solutions sont utilisables en pratique, et n'altèrent même pas la rapidité des routeurs pour ASIC.

1.5.2.2 FPGA

Les contraintes sont plus délicates à poser, surtout dans les FPGA Altera. Laurent Sauvage [401] a montré comment contraindre le placement par l'appariement de deux portes dans un CLB (instance reconfigurable dite « *Compound Logic Block* ») qui dispose en l'espèce de deux sorties indépendantes. Néanmoins des solutions intéressantes existent pour les FPGA Xilinx. Par exemple :

- Dans DWDDL (Double WDDL [479]), le module complet est copié-collé à un autre endroit, et dualisé. Cela revient à remplacer une fonctionnalité $f(\cdot)$ par $\neg f(\neg \cdot)$. Globalement, cette duplication permet de cacher une évaluation anticipée par une évaluation non-anticipée (ou vice-versa). Cette solution reste tout de même un pis-aller.
- Dans [465], une stratégie de réservation de ressources de placement et de routage semblable à celle de la *backend duplication* [191] est présentée. La netlist DPL placée et routée est obtenue par la translation d'une moitié, ce qui est possible grâce à une opération de copier-coller.

En revanche, en FPGA, si les contraintes sont moins faciles à gérer, on peut jouer sur d'autres facteurs. Nos études ont montré que les solutions suivantes permettent d'améliorer la sécurité :

- Utilisation de logiques sans évaluation précoce [37] : la logique présentée dans ce papier, appelée « *WDDL w/o EE* », s'appuie sur des portes à deux entrées qui n'évaluent VALID que si les entrées sont VALID.
- Placement contraint dans une ressource commune (sans contrainte de routage) [402] : un appariement au niveau porte est forcé grâce à un regroupement deux instances vraie et fausse au sein d'un même CLB.
- Réduction des interconnexions (nombre total de nœuds) en général et de la sortance (nombre de branches des fourches au sortir des portes logiques, *aka fanout*) en particulier [42] : les déséquilibres de routage sont en effet réduits si la quantité de routage est elle-même réduite.
- Correction manuelle après caractérisation par analyse stochastique [40] : cette technique se justifie dès lors que très peu de paires différentielles sont déséquilibrées. Or ceci est le cas en pratique, car les algorithmes de routage (comme `A*`) n'ont pas de raison de traiter différemment les deux équipotentielles d'une paire.

1.5.3 Conclusions

Par rapport au masquage, qui s'appuie sur de l'aléa, la dissimulation d'information sous-entend un équilibrage. Ainsi, la spécificité des CM de type DPL est de nécessiter et une structure logique et une implémentation physique sans biais. C'est pour cette raison que les logiques DPL s'implémentent plus facilement en matériel qu'en logiciel, car le concepteur a une plus grande connaissance et éventuellement une plus grande maîtrise des aspects physiques. Pour autant, le DPL logiciel n'est pas illusoire, comme le montre la preuve de concept illustrée dans le travail [220].

Il est souvent reproché aux DPL d'être délicates à implémenter, à cause des contraintes

à imposer sur les étapes de P&R. Néanmoins, il est utile de rappeler que ces logiques peuvent être implémentées depuis le code source par synthèse logique, profitant ainsi de toutes les optimisations de haut niveau offertes par les outils de CAO. Par exemple, dans l'article [205], nous montrons comment la capacité d'inférence des synthétiseurs logiques permet d'instancier de façon particulièrement appropriée des primitives pour des logiques DPL. Le masquage ne permet généralement pas de jouir d'un tel confort, car l'ordre d'utilisation des masques est primordial. Ainsi, les contremesures de masquage logiciel sont-elles bien souvent écrites manuellement en langage assembleur.

1.6 Dissimulation : évaluation

Les CM de type DPL présentent à la fois une résistance aux attaques actives et passives. Le premier niveau de test est une vérification bit par bit (Sec. 1.6.1). Des attaques passives plus évoluées, d'ordres élevés, sont également envisageables (Sec. 1.6.2). Vis-à-vis des attaques en perturbation, les DPL présentent une propriété intéressante de résilience (Sec. 1.6.3).

1.6.1 Analyse de la fuite : variance

L'évaluation des CM de dissimulation diffère de celle des CM de masquage, car l'on ne connaît cette fois-ci pas la fuite. Plus précisément, le modèle théorique de la fuite est une constante. La stratégie d'analyse de fuite est donc nécessairement aveugle : il s'agit de scruter toutes les variables sensibles. Comme en DPL, l'activité est sensée être constante, une simple analyse de la variance (non conditionnelle) de chaque nœud suffit. Elle permet de repérer les possibles variables qui fuient, à cause soit de la présence d'un *glitch* dépendant des données portées par la ressource, d'une date d'activité variable ou d'un déséquilibre du routage. Cette approche a notamment été mise en œuvre dans [210], disponible sous sa forme longue dans la partie E à la page 155.

1.6.1.1 Pertinence de la variance inconditionnelle

Ce paragraphe justifie que la variance est une métrique de fuite dans les logiques DPL. Pour modéliser la fuite, nous notons :

- X, le canal caché (*i.e.* une mesure physique prise subrepticement pendant un calcul), et
- Y, la variable sensible (*i.e.* un vecteur de bits que l'attaquant suppose être utilisé pendant le même calcul),

toutes deux vues comme des variables aléatoires. L'hypothèse d'une exploitabilité en analyse de canaux auxiliaires est que X dépend de Y. Ainsi, l'attaquant emploie-t-il un distingueur pour tester la dépendance entre X et Y. Par exemple, il peut recourir à la « covariance » entre X et Y, notée $\mathsf{Cov}[X, Y]$. En pratique, l'attaquant ne connaît pas Y mais peut deviner sa valeur moyennant des hypothèses sur la clé secrète. Lorsque cette clé est mal devinée, le distingueur reste alors proche de zéro. C'est de cette façon que l'on distinguera la bonne hypothèse de clé des mauvaises. Ainsi, une attaque sera

d'autant plus facile que le contraste entre la valeur du distingueur pour la bonne clé et les mauvaises est important. Donc, une métrique de fuite pertinente est simplement la valeur du distingueur pour la bonne clé. Pour l'analyse, supposons que la fuite X se décompose en la somme :

- d'une partie dépendant de la variable sensible Y (*e.g.* un changement de bit, *aka bit-flip*), et
- d'un bruit indépendant, noté N, qui modélise l'activité des autres nœuds du circuit, c'est-à-dire le bruit algorithmique, plus le bruit de mesure (bruit de grenaille, de quantification, *etc.*).

Par conséquent, $\mathsf{Cov}[X,Y] = \mathsf{Cov}[Y+N,Y] = \mathsf{Cov}[Y,Y] + \cancel{\mathsf{Cov}[N,Y]} = \mathsf{V}[Y]$. De plus, comme Y et N sont indépendants, $\mathsf{V}[X] = \mathsf{V}[Y] + \mathsf{V}[N]$. Or $\mathsf{V}[N]$ est indépendant de l'hypothèse sur la clé. D'où la pertinence de l'usage de la variance $\mathsf{V}[X]$ comme une métrique de fuite sur les canaux auxiliaires. Intuitivement, elle reflète dans quelle mesure la fuite est présente dans les mesures. Si $\mathsf{V}[X] = 0$, les courbes sont constantes ; il n'y a rien que l'on puisse apprendre d'elles. Si $\mathsf{V}[X]$ est non nul, on peut s'attendre à ce qu'une partie des variations soit due à la variable sensible ; plus $\mathsf{V}[X]$ est grand, plus l'est également la dépendance avec Y.

1.6.1.2 La différence de moyennes est une covariance

On peut remarquer que :

$$\mathsf{Cov}[X,Y] = \mathsf{E}[(X - \mathsf{E}[X]) \times (Y - \mathsf{E}[Y])] = \mathsf{E}[X \times (Y - \mathsf{E}[Y])]\,.$$

Donc :

$$\begin{aligned}\mathsf{Cov}[X,Y] &= \sum_x \sum_y \mathsf{P}[X = x \wedge Y = y] \times x \times (y - \mathsf{E}[Y]) \\ &= \sum_x \sum_y \mathsf{P}[X = x \mid Y = y] \times \mathsf{P}[Y = y] \times x \times (y - \mathsf{E}[Y])\,. \qquad (1.7)\end{aligned}$$

Quand Y est une variable booléenne, $Y \in \{0,1\}$. Nous nous restreignons à ce cas dans les lignes ci-dessous. En cryptographie, les variables internes des algorithmes sont équilibrées (*i.e.* $\mathsf{P}[Y=0] = \mathsf{P}[Y=1] = 1/2$), d'où $Y - \mathsf{E}[Y] \sim \mathcal{U}(\{-1/2, +1/2\})$. Dans ce cas, l'Eqn. (1.7) se réécrit :

$$\begin{aligned}\mathsf{Cov}[X,Y] &= \sum_x \mathsf{P}[X = x \mid Y = 0] \times x \times \left(-\frac{1}{2}\right) + \sum_x \mathsf{P}[X = x \mid Y = 1] \times x \times \left(+\frac{1}{2}\right) \\ &= \frac{1}{2}\Big(\mathsf{E}[X \mid Y = 1] - \mathsf{E}[X \mid Y = 0]\Big)\,.\end{aligned}$$

C'est le même test mono-bit (à un facteur $1/2$ près qui est sans conséquence aucune) que la différence de moyennes de P. C. Kocher *et al.* [248] (voir également le raisonnement fait au §2.1.1 de [208]).

De même, on peut constater que les métriques M1, M2 et M3 de [210] (voir partie E à la page 155) sont des variantes de la variance, qui considèrent différentes façons de résumer

l'aspect vectoriel des mesures en un scalaire. Par ailleurs, la métrique PAT (*Power Attack Tolerance*), introduite dans [267], est homogène à l'inverse d'une variance.

1.6.1.3 Lien avec la théorie de l'information

Le papier [434] de François-Xavier Standaert *et al.* encourage l'usage de l'information mutuelle, $\mathsf{I}[X;Y]$, comme une métrique de fuite. Dans le cas DPL, les deux métriques $\mathsf{V}[X]$ et $\mathsf{I}[X;Y]$ sont quasiment équivalentes. Pour le démontrer, nous rappelons que $\mathsf{I}[X;Y] = \mathsf{H}[X] - \mathsf{H}[X \mid Y]$, et que $\mathsf{H}[G] = \frac{1}{2}\log_2\left(2\pi e \mathsf{V}[G]\right)$ bit si $G \sim \mathcal{N}(\mathsf{E}[G], \mathsf{V}[G])$ est une variable aléatoire qui suit une loi normale (*cf.* Eqn. (1.11)). En effet, en supposant que $N \sim \mathcal{N}(0, \sigma^2)$ est un bruit gaussien et que $\mathsf{V}[N] = \sigma^2 \gg \mathsf{V}[Y]$ (les mesures sont très bruitées, *i.e.* la SPA, ou *Simple Power Analysis* [248], est impossible), nous avons :

- $\mathsf{H}[X] = \frac{1}{2}\log_2\left(2\pi e(\sigma^2 + \mathsf{V}[Y])\right)$, si l'on suppose que la mixture de gaussiennes $N + Y$ est en fait bien approximée par une gaussienne, de variance $\mathsf{V}[N + Y] = \mathsf{V}[N] + \mathsf{V}[Y]$ par indépendance ;
- $\mathsf{H}[X \mid Y] = \frac{1}{2}\log_2\left(2\pi e\sigma^2\right)$, car sans bruit, $X \mid Y = y$ est déterministe (et vaut y), et avec bruit, $X \mid Y = y \sim \mathcal{N}(y, \sigma^2)$.

Maintenant, $\log_2\left(2\pi e(\sigma^2 + \mathsf{V}[Y])\right) = \log_2\left(2\pi e\sigma^2 \times (1 + \mathsf{V}[Y]/\sigma^2)\right) = \log_2\left(2\pi e\sigma^2\right) + \frac{1}{\ln 2}\frac{\mathsf{V}[Y]}{\sigma^2} + \mathcal{O}\left(\frac{\mathsf{V}[Y]}{\sigma^2}\right)$, grâce à l'application du développement limité : $\ln(1+\epsilon) = \epsilon + \mathcal{O}(\epsilon)$, valide si $\epsilon \longrightarrow 0$. Ainsi, au premier ordre en $\frac{\mathsf{V}[Y]}{\sigma^2} \ll 1$, $\mathsf{I}[X;Y] = \frac{1}{2\ln 2}\frac{\mathsf{V}[Y]}{\sigma^2}$. Donc $\mathsf{I}[X;Y] \propto \mathsf{V}[Y] = \mathsf{V}[X] - \sigma^2$, CQFD. Ce n'est donc pas étonnant que l'on ait dans [210] un accord (en terme d'ordre de classement des différents modules) entre les métriques de variance et d'information mutuelle (voir les courbes de [210, Fig. 16]).

1.6.1.4 Avertissement

L'usage de la variance (ou de l'écart-type) des courbes pour comparer des CM n'est pas toujours approprié. Effectivement, on peut imaginer des situations où la variance est grande mais la fuite faible. Par exemple, cela peut être le cas d'un circuit DPL (pour lequel la variance de Y est nulle si l'implémentation est correcte), où l'on aurait artificiellement ajouté du bruit à dessein pour augmenter le bruit environnemental. Également, la variance de Y peut être très grande, comme dans les schémas de masquage (qui emploient des masques grandement entropiques), mais l'exploitation peut s'avérer difficile en pratique (des attaques d'ordre élevé sont nécessaires). Néanmoins, dans un contexte où la CM est connue (tel [210]), il n'y a ni sources de bruit artificielles ni de masques.

La présence de la précharge à NULL permet de simplifier les modèles de fuite de la distance de Hamming au poids de Hamming. On pourrait argumenter que cette propriété est à l'avantage de l'attaquant. C'est effectivement vrai, sauf si le concepteur s'en empare également pour tester de façon unitaire le déséquilibre de chaque ressource, et le corriger le cas échéant.

1.6.2 Caractérisation de la fuite et attaques passives

L'approche mono-bit donne une idée de la fuite au premier ordre. Si l'implémentation physique est soignée, elle ne devrait pas exister. Mais il est possible qu'il subsiste des fuites aux ordres plus élevés, *i.e.* qui impliquent plusieurs bits. Il y a au moins deux raisons de nature physique pour expliquer de telles fuites :

1. Les couplages capacitifs entre les fils induisent des phénomènes de diaphonie. Par exemple, un fil dit agresseur peut influencer la propagation d'un front sur un fil voisin dit victime si les deux ont une transition. Cet événement singulier n'advient qu'à la condition d'avoir conjointement deux transitions sur deux fils de paires différentielles différentes. La diaphonie induit donc une fuite conditionnelle à un « et logique » entre deux variables (fuite d'ordre deux).
2. Le phénomène de propagation anticipée est d'autant plus fort qu'il se manifeste profondément dans la logique, car les différences de délais ont des sources plus variées et surtout s'ajoutent le long du chemin. Or, les variables proches de la sortie des cônes de logique dépendent de nombreux bits d'entrée, et donc ne peuvent être exploitées qu'avec des modèles d'ordres particulièrement élevés.

Le modèle de fuite d'ordre élevé peut être mis en évidence par une analyse stochastique. Nous avons constaté sur des implémentations de DPL non équilibrées qu'il existe à la fois des fuites d'ordre un et d'ordre élevé [120, 41]. Dans l'étude [120], nous montrons que les fuites peuvent correspondre à des modèles non-intuitifs, mais que tous sont exprimables comme des polynômes dans les variables intermédiaires (aussi bien les couplages en DPL que les *glitches* en CM de masquage). Le travail [41] explicite quant à lui que la fuite est d'autant plus importante que le modèle de fuite est d'ordre élevé, et apporte une validation pratique à cette observation. Ces résultats sont illustrés dans la Fig. 1.15, pour une variable Y encodée sur $n = 8$ bits (*i.e.* $Y \in \mathcal{Y} = \mathbb{F}_2^8$).

Différents autres travaux de recherches mentionnent également ces fuites [390, 440].

1.6.3 Résilience aux fautes

On peut distinguer deux types de fautes transitoires : les fautes asymétriques, où l'erreur ne peut amener que vers une valeur privilégiée (par exemple '0'), et les fautes symétriques, où l'on peut avoir des *bit-flips* dans les deux sens ($0 \rightarrow 1$ et $1 \rightarrow 0$), éventuellement avec des probabilités différentes. Sur les circuits DPL avec espaceur à '00', toute perturbation qui se traduit par une violation de temps de prépositionnement (*setup time violation*) conduit à une faute asymétrique vers zéro. Effectivement, en phase d'évaluation, quand les valeurs sont divulguées à l'extérieur, la *netlist* part d'un état tout à zéro. Donc, si des signaux sont ralentis, la valeur qui sera échantillonnée sera zéro au lieu de la valeur valide. On peut noter qu'il peut également y avoir une faute semblable en phase de précharge. La manifestation sera alors opposée : un fil qui était sensé passer à zéro est trop lent et donc reste à sa valeur un. Ici, il s'agit donc d'une faute asymétrique à un. Mais comme elle se produit durant une phase où les données ne sortent pas, elle n'est pas exploitable. D'ailleurs, elle concerne une donnée non sensible (la valeur

$$X(Y) = \sum_{i=1}^{n} \beta_i \cdot \underbrace{\left(Y_i - \frac{1}{2}\right)}_{\text{Base du 1er ordre}} + \sum_{i \neq j} \beta_{i,j} \cdot \underbrace{\left(Y_i - \frac{1}{2}\right) \cdot \left(Y_j - \frac{1}{2}\right)}_{\text{Base du 2nd ordre}} .$$

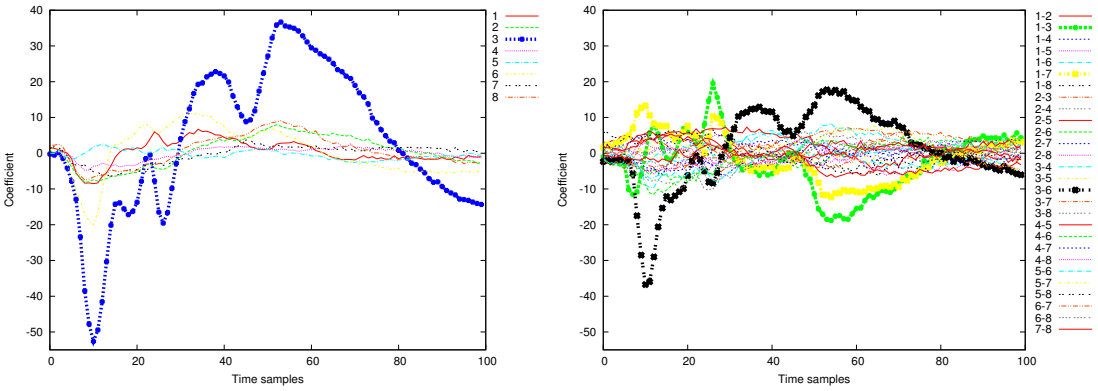


FIGURE 1.15 – Caractérisation stochastique d'une sbox d'un circuit AES en DPL non-équilibré. Premier ordre à gauche (coefficients β_i), second à droite (coefficients $\beta_{i,j}$).

NULL). De telles fautes asymétriques sont typiquement causées soit par une décélération des signaux, dues typiquement à des perturbations globales (*glitches* sur l'alimentation, sous-alimentation durable, chauffage du composant, *etc.*), soit par une accélération de l'horloge (*overclocking*, *etc.*). Les fautes symétriques nécessitent un moyen d'action local. Effectivement, pour avoir l'opportunité de fauter soit vers '0' soit vers '1', il faut pouvoir rendre passant soit un transistor de type N, soit de type P. Ceci s'obtient par exemple par un tir laser localisé, à même d'être à l'origine de la création de charges libres dans les parties actives du silicium.

Les circuits DPL sont résilients aux fautes symétriques, car une fois un fil à un effacé en un zéro, il est impossible à l'attaquant de savoir si la valeur valide non fautée était '0' ou '1'. Donc le chiffré erroné n'est pas sensible, car ne dépendant pas de la vraie valeur secrète. C'est en ce sens que les circuits DPL (WDDL [411] et autres) sont résilients aux fautes asymétriques.

Maintenant, en présence de fautes asymétriques, il est possible de fauter les deux fils portant une valeur de façon antinomique, créant ainsi une inversion logique cohérente. Ce type de faute amènera assurément à une attaque en faute réussie. Cependant, si l'on se place dans un contexte où l'endroit d'apparition de la faute est aléatoire, alors la probabilité de fauter les deux fils d'une même paire décroît. Si de plus la logique est résistante à la propagation anticipée, elle propagera une valeur nulle même si la faute n'a pas eu d'impact. Ainsi, en cas d'injection de fautes multiples, même si une faute valide apparaît, il est fort probable qu'elle se fasse absorber par une vague de NULL [33].

Récemment, une attaque à la frontière entre les canaux cachés et les fautes a été publiée. Il s'agit de la *Fault Sensitivity Analysis* (FSA [266]). Le canal caché est le ni-

veau de stress nécessaire minimal pour faire apparaître une faute. Il se trouve que cette valeur dépend potentiellement d'une variable sensible. La FSA appliquée sur les bits individuels d'une implémentation peut fonctionner, notamment s'ils souffrent de propagation anticipée. Maintenant, même si les bits évaluent à vrai et à faux à une date identique, les différences de délais entre bits peuvent être exploitées [264]. De plus, grâce à une autre technique (non documentée), une autre équipe a également réussi (avec succès quoiqu'avec de très nombreuses interactions avec le système) une FSA sur une implémentation DPL [322].

1.6.4 Conclusion

En fonction de leur sophistication (et donc de leur coût), les circuits implémentés en DPL présentent une résistance de bonne à excellente aux attaques passives. Une contremesure DPL parfaite est à la fois dépourvue de *glitches*, de propagation anticipée, a un routage équilibrée et des paires de fils capacitivement découplées entre elles. Toutes ces qualités peuvent être réunies simultanément, comme nous en avons fait le preuve dans [175, 210]. Par ailleurs, même avec un niveau de sécurité inférieur (pour des raison de compromis sécurité / coût), il est intéressant de noter que s'il subsiste une fuite, alors elle ne concernera de toutes façons que quelques octets sensibles, si bien qu'un attaquant ne sera pas en mesure de conclure une attaque. Le même raisonnement ne s'appliquerait pas au masquage : si l'on sait monter une attaque d'ordre élevé (qui exploite la distribution des masques), alors on retrouvera de façon consistante tous les octets de clé.

Une autre propriété qui rend les DPL attrayantes par rapport au masquage est leur immunité aux attaques actives. Certes, un attaquant plus puissant, qui maîtrise bien son injection, aura davantage de chance de réussir tout de même une attaque en injection de faute [471].

1.7 Résilience

Comme illustré dans les sections précédentes, la sécurité physique peut trouver ses racines dans un matériel rendu robuste, indépendemment de son usage. Une alternative est de relâcher les contraintes sur le matériel, et les reporter sur l'interface. En interdisant à l'utilisateur certaines manipulations, des attaques peuvent être neutralisées. Nous montrons ici comment résister :

- aux attaques en observation grâce à une mise à jour fréquente du secret,
- aux attaques en perturbation en empêchant l'attaquant de choisir son clair, et
- aux attaques en déroutement grâce à un chaînage protocolaire.

1.7.1 Résilience contre les attaques en observation

L'usage de secrets éphémères permet de contrer des attaques en observation d'un secret sensé être constant. Différentes techniques de diversification fréquente de clés ont été proposées. Elles se reposent sur deux principes :

– La recherche dans un annuaire d'une clé précalculée (baptisé "*Indexed Key Update*" [245], *aka* IKU) ;
– Le calcul d'une clé dérivée dynamiquement (baptisé "*Fresh-ReKeying*" [304], *aka* FRK).

Un inconvénient d'IKU est que le nombre de clés disponibles est fini, car prédéterminé statiquement. De plus, il s'agit d'un protocole à état, qui nécessite donc de disposer de mémoire non-volatile. De son côté, FRK peut idéalement générer toutes les clés admissibles par l'algorithme sous-jacent, et ce à la volée (la dérivation de la clé utilise une nonce dynamique). Cependant, FRK peut être victime d'une attaque sur l'algorithme de dérivation de clé, qui lui utilise la clé secrète. Toute la difficulté est donc d'inventer des primitives légères (moins coûteuses que l'algorithme à protéger et rapides) pour la dérivation de clé.

1.7.2 Résilience contre les attaques en perturbation

Les attaques en perturbation fonctionnent généralement par comparaison, en analysant les différences entre un couple de cryptogrammes correct et fauté. Ceci est vrai évidemment pour des DFA, mais aussi de certaines « *safe errors* » [477], en l'occurrence celles où le résultat est sorti sans être testé. Effectivement, pour pouvoir décider si un calcul est correct ou non, il faut pouvoir disposer d'une référence. La résilience contre les attaques en perturbation fonctionne donc de la façon suivante : on ne laisse jamais l'occasion à l'attaquant de prédire ce qui s'est passé. Cette approche s'appuie sur l'utilisation d'aléa dès le début des protocoles, avec comme règle d'or que la partie le plus faible commence la transaction. Il est possible de combiner les techniques de résilience aux attaques en observation et en perturbation ; c'est notamment le sujet de notre article [209].

1.7.3 Résilience contre les attaques sur les protocoles

Certains protocoles consistent en différentes étapes, qui permettent chacune à un utilisateur légitime de gagner des droits. Un exemple typique est une authentification mutuelle autorisant une étape d'échange de secret permettant d'initier une communication confidentielle. Les deux méthodes présentées dans les Sec. 1.7.1 et 1.7.2 s'appliquent à l'intérieur de chaque phase, alors que la CM que nous recherchons dans cette section concerne l'intégrité du flot d'exécution. Une voie d'attaque classique est alors de sauter une ou plusieurs étapes grâce à une injection de faute, par exemple. Dans les implémentations traditionnelles des protocoles, les étapes sont indépendantes, ce qui permet effectivement de gagner des privilèges indûment en contournant des étapes.

Maintenant, la résilience au niveau protocolaire consiste à rendre les étapes à la fois séquentielles mais en plus dépendantes de façon calculatoire les unes des autres [179]. Ainsi, en sauter une ou plusieurs n'apporte aucun avantage à l'attaquant, car une partie d'un « état cryptographique » inconnu n'est pas disponible à l'attaquant. De ce fait, le déroulement du protocole est erroné si l'attaquant essaye de poursuivre malgré l'altération de la séquence nominale des étapes.

1.8 Conclusion et perspectives

Les contremesures de masquage et de dissimulation permettent de rendre plus compliquées les attaques sur les implémentations de chiffrement symétrique. Elles sont aussi toutes deux aisément implémentables (et ce de façon automatisable) dans des flots ASIC ou FPGA, avec, il est vrai, différents niveaux d'expertise requis selon la CM concernée. Les limites du masquage s'étudient grâce à des outils de statistique, en analysant des distributions de probabilités. L'outil maître pour évaluer les imperfections des logiques DPL est l'analyse stochastique, qui tente de modéliser des fuites combinant plusieurs bits. L'inconvénient du masquage est que les attaques sont structurelles : si une attaque réussit sur une partie de la clé (un octet), alors *a priori* tous les autres octets sont de façon consistante vulnérables à la même attaque. La situation est différente avec les DPL : en cas de problème d'implémentation, seuls les octets de clés impliqués dans les parties déséquilibrées sont compromis, et non toute la clé. Une façon encore moins coûteuse de protéger les implémentations cryptographiques contre les attaques physique est la résilience. C'est dans l'usage de primitives *a priori* non protégées que l'on arrive à protéger les secrets. L'avantage des approches résilientes est leur simplicité de mise en œuvre et (idéalement), leur prouvabilité. L'inconvénient est que les contraintes d'usage ne sont souvent pas compatibles avec les standards actuels. Ainsi, nous pensons que davantage de recherche dans ce domaine pourrait globalement être profitable à l'industrie de la sécurité de systèmes embarqués.

En termes de perspectives, il est important de souligner qu'il reste beaucoup de pistes d'amélioration. Quelques grands défis à relever sont listés ci-après.

- Formaliser plus précisément les gains de sécurité et les diminutions de fuite que l'on peut obtenir en utilisant au mieux l'entropie de nombres aléatoires dédiés au masquage (technique du *leakage squeezing* [279], fonction alpha [94, 282] ou masquage du premier ordre parfait, restriction à un sous-ensemble des masques [332, 334]), et quantifier les risques.
- Développer des méthodes automatisables de conception guidée par la sécurité, et les outils (logiciels d'analyse de fuite et/ou d'inférence de modèle, et plateformes expérimentales) de vérification associés. Dans le monde de la carte à puce, d'aucuns estiment [6] que le marché est trop petit pour ce type de commodités ; mais en sécurité en général, il y a de la place pour le développement de ces méthodes.
- Mieux comparer les différentes options de protections, dans un cadre unique ou au contraire en exhibant une association entre condition d'usage et CM associée. Un travail de comparaison entre masquage et DPL a déjà été initié dans [299]. Cependant cette analyse est basée sur des attaques et non des métriques de fuite, et est heuristique car réalisée sur des échantillons de traces expérimentales (non représentatives).
- Favoriser l'utilisation de méthodes formelles (aussi bien de la vérification de propriétés que de la preuve de théorèmes) dans les flots de conception de composants

6. Voir par exemple le *post* d'Éric Vétillard, daté du 30 septembre 2011, sur son blog en ligne : http://javacard.vetilles.com/2011/09/30/my-last-day-at-trusted-logic/.

de sécurité [181].
- Permettre le relâchement des contraintes pesant sur l'implémentation par différentes voies (cryptographie légère, protocoles dits « résilients » [206, 209, 179]). Étudier de telles solutions et les promouvoir comme des standards.

1.9 Annexe : notations et résultats fondamentaux

1.9.1 Notations

Les variables aléatoires sont notées en majuscules (*e.g.* X) et leurs réalisations en minuscules (*e.g.* x), tandis que leur espace de définition est représenté par une lettre calligraphiée (*e.g.* $\mathcal{X}$). Par exemple, l'espace de définition du bruit N est noté $\mathcal{N}$, symbole qui sert aussi à définir la loi normale de variance μ et de variance σ^2 : $N \sim \mathcal{N}(\mu, \sigma^2)$. Nous supposerons les variables aléatoires scalaires. La probabilité d'une variable aléatoire X en $x \in \mathcal{X}$ est notée $\mathsf{P}[X = x]$, ou simplement $\mathsf{P}[x]$ quand il est clair que l'on s'intéresse à X.

1.9.1.1 Moments

Les deux premiers moments de X sont :
- Son espérance, notée $\mathsf{E}[X]$, est définie comme $\sum_{x \in \mathcal{X}} x \cdot \mathsf{P}[X = x]$. Si X n'est pas discrète mais continue, alors $\mathsf{E}[X] \doteq \int_{x \in \mathcal{X}} x \cdot \mathsf{P}[X = x] \, \mathrm{d}x$, où cette fois-ci $\mathsf{P}[X = x]$ est la densité de probabilité de X. Dans les autres définitions à venir, nous ne faisons plus la différence entre ces deux cas : le lecteur comprendra qu'il faut interpréter l'expression comme une somme arithmétique si X est discrète ou comme une intégrale continue autrement (typiquement quand un bruit gaussien est ajouté à X), en fonction du contexte.
- Sa variance, notée $\mathsf{V}[X]$, est un nombre réel positif[7] défini comme :

$$\mathsf{V}[X] \doteq \mathsf{E}[(X - \mathsf{E}[X])^2] \ .$$

Proposition 1. $\mathsf{V}[X] = \mathsf{E}[X^2] - (\mathsf{E}[X])^2$.
Cette propriété permet d'estimer une variance en ligne à l'aide de deux « accumulateurs », qui somment respectivement le carré de tirages de x de X et les x simplement.

Étant donnée une autre variable aléatoire Y, nous introduisons la nouvelle variable aléatoire $\mathsf{E}[X \mid Y]$. Il s'agit d'une fonction de Y, qui prend la valeur $\sum_x x \cdot \mathsf{P}[X = x \mid Y = y]$ avec la probabilité $\mathsf{P}[Y = y]$ en y[8].

Par la loi des espérances totales, nous avons : $\mathsf{E}[\mathsf{E}[X \mid Y]] = \mathsf{E}[X]$. En effet, $\mathsf{E}[\mathsf{E}[X \mid Y]] = \sum_y \mathsf{P}[Y = y] \cdot \sum_x x \cdot \mathsf{P}[X = x \mid Y = y] = \sum_x x \cdot \sum_y \mathsf{P}[Y = y] \cdot \mathsf{P}[X = x \mid Y = y] = \sum_x x \cdot \sum_y \mathsf{P}[X = x \wedge Y = y] = \sum_x x \cdot \mathsf{P}[X = x] = \mathsf{E}[X]$.

7. $\mathsf{V}[X]$ est homogène au carré de X ; cela signifie que si X a comme unité u (*e.g.* u est un micro-volt, *i.e.* $u = \mu V$), alors $\mathsf{V}[X]$ s'exprime en unité u^2 (*e.g.* $u^2 = \mu V^2$).

8. Cette définition et cette notation sont usuelles, et notamment utilisées dans d'autres papiers, comme [365, §3.3].

De façon similaire à $\mathsf{E}[X \mid Y]$, $\mathsf{V}[X \mid Y]$ est définie comme $\mathsf{E}[X^2 \mid Y] - (\mathsf{E}[X \mid Y])^2$. Il s'agit d'une variable aléatoire, qui dépend de Y mais pas de X. On appelle classiquement les valeurs de cette variable aléatoire $\mathsf{V}[X \mid Y = y]$ les variances conditionnelles en $Y = y$.

La variable aléatoire X peut être classifiée en fonction des valeurs prises par Y.

– On définit la variance intra-classes comme $\mathsf{E}[\mathsf{V}[X \mid Y]]$, alors que
– la variance inter-classes se définit comme $\mathsf{V}[\mathsf{E}[X \mid Y]]$.

Le théorème de l'analyse de variance énonce que la variance totale se décompose exactement en la somme des variances intra- et inter-classes. Formellement, cela signifie que :

Proposition 2. *Analyse de variance.* $\mathsf{V}[X] = \mathsf{E}[\mathsf{V}[X \mid Y]] + \mathsf{V}[\mathsf{E}[X \mid Y]]$.

Démonstration.

$$\begin{aligned} & \mathsf{E}[\mathsf{V}[X \mid Y]] + \mathsf{V}[\mathsf{E}[X \mid Y]] \\ = \; & \mathsf{E}[\mathsf{E}[X^2 \mid Y] - (\mathsf{E}[X \mid Y])^2] + \mathsf{E}[(\mathsf{E}[X \mid Y])^2] - (\mathsf{E}[\mathsf{E}[X \mid Y]])^2 \\ = \; & \mathsf{E}[\mathsf{E}[X^2 \mid Y]] - \cancel{\mathsf{E}[(\mathsf{E}[X \mid Y])^2]} + \cancel{\mathsf{E}[(\mathsf{E}[X \mid Y])^2]} - (\mathsf{E}[\mathsf{E}[X \mid Y]])^2 \\ = \; & \mathsf{E}[X^2] - (\mathsf{E}[X])^2 = \mathsf{V}[X] \ . \end{aligned}$$

□

1.9.1.2 Entropies

L'entropie d'une variable aléatoire X est égale à $\mathsf{H}[X] \doteq -\sum_{x \in \mathcal{X}} \mathsf{P}[X = x] \cdot \log \mathsf{P}[X = x]$. Par convention, si pour un $x \in \mathcal{X}$, $\mathsf{P}[X = x] = 0$, alors $\mathsf{P}[X = x] \cdot \log \mathsf{P}[X = x] = 0$. Cela correspond à la valeur de $\lim_{\epsilon \longrightarrow 0^+} \epsilon \cdot \log \epsilon$. De la base du logarithme dépend l'unité de l'entropie. Lorsque la base :

– est e, on notera $\log_e = \ln$ (pour logarithme néperien, ou naturel) et l'on dira que l'entropie s'exprime en nats ;
– est 2, on notera $\log_2$ et l'on dira que l'entropie est en bits (abréviation de *binary units*, *i.e.* unités binaires).

Sauf indication contraire, toutes les entropies calculées dans ce rapport sont données en bits. Par la suite, on rencontrera des termes $\frac{1}{\ln 2}$ d'ajustement pour les unités en bits ; notons que $\frac{1}{\ln 2} = \log_2(e)$.

L'entropie conditionnelle de X sachant Y est égale à : $\mathsf{H}[X \mid Y] \doteq \sum_{y \in \mathcal{Y}} \mathsf{P}[Y = y] \cdot \mathsf{H}[X \mid Y = y]$. Cette notion permet de définir l'information mutuelle entre X et Y, comme : $\mathsf{I}[X;Y] \doteq \mathsf{H}[X] - \mathsf{H}[X \mid Y]$.

Ces notions sont reliées par différentes relations de conservation, représentées traditionnellement par un diagramme de Venn. Il est illustré dans la Fig. 1.16.

L'entropie croisée de deux variables aléatoires X et Y définies sur le même domaine $\mathcal{Z}$ se note $\mathsf{H}[X,Y]$, et est définie par : $\mathsf{H}[X,Y] \doteq -\sum_{z \in \mathcal{Z}} \mathsf{P}[X = z] \cdot \log_2 \mathsf{P}[Y = z]$.

Cette notion est utile pour définir une métrique qui quantifie la dissemblance des distributions de deux variables aléatoires X et Y, appelée divergence de Kullback-Leibler et notée $\mathsf{D}_{\mathsf{KL}}[X \parallel Y] \doteq \sum_{z \in \mathcal{Z}} \mathsf{P}[X = z] \cdot \log_2 \frac{\mathsf{P}[X=z]}{\mathsf{P}[Y=z]}$. On a alors aussi : $\mathsf{D}_{\mathsf{KL}}[X \parallel Y] = \mathsf{H}[X,Y] - \mathsf{H}[X]$.

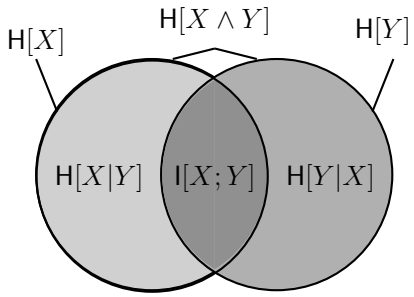


FIGURE 1.16 – Diagramme d'information de Venn.

1.9.2 Théorie de l'information avec des variables normales

Soit X une variable aléatoire normale de moyenne μ, de variance σ^2. On note $X \sim \mathcal{N}(\mu, \sigma^2)$. La densité de probabilité de la variable aléatoire X au point $x \in \mathbb{R}$ est égale à $\phi_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2$. Ici, la fonction exponentielle est à comprendre en base e ; c'est-à-dire que $\forall \xi \in \mathbb{R}, \exp \xi = e^{\xi}$.

1.9.2.1 Moments

Le moment d'ordre zéro d'une loi normale est égale à l'aire de sa densité de probabilité :

$$\forall(\mu, \sigma), \quad \int_{\mathbb{R}} \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x = 1\,. \tag{1.8}$$

In s'agit d'un résultat classique, appelé communément intégrale de Gauss.

Pour évaluer le moment du premier ordre, il faut calculer :

$$\begin{aligned}
\int_{\mathbb{R}} (x-\mu) \times \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &= \\
\int_{\mathbb{R}} -\sigma^2 \frac{\partial}{\partial x}\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \times \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &= \\
-\sigma^2 \int_{\mathbb{R}} \frac{\partial}{\partial x}\phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &= \\
-\sigma^2 \Big[\phi_{\mu,\sigma^2}(x)\Big]_{-\infty}^{+\infty} &= 0\,,
\end{aligned}$$

d'où

$$\int_{\mathbb{R}} x \times \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x = \mu\,. \tag{1.9}$$

Maintenant, pour le moment du second ordre, il s'agit d'utiliser une intégration par

parties pour calculer :

$$\begin{aligned}
\int_{\mathbb{R}} (x-\mu)^2 \times \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &= \\
-\sigma^2 \int_{\mathbb{R}} \underbrace{(x-\mu)}_{u} \times \underbrace{\frac{\partial}{\partial x}\phi_{\mu,\sigma^2}(x)}_{v'}\,\mathrm{d}x &= \\
-\sigma^2 \cancel{\Big[u \times v\Big]_{-\infty}^{+\infty}} - -\sigma^2 \int_{\mathbb{R}} u' \times v\,\mathrm{d}x &= \\
+\sigma^2 \int_{\mathbb{R}} \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &= \sigma^2\,,
\end{aligned}$$

d'où

$$\begin{aligned}
\int_{\mathbb{R}} x^2 \times \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &= \text{// Commentaire : } \left\{ \begin{array}{rcl} x^2 &=& ((x-\mu)+(\mu))^2 \\ &=& (x-\mu)^2 + \mu^2 - 2\mu \times (x-\mu) \end{array} \right. \\
\int_{\mathbb{R}} (x-\mu)^2 \times \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &+ \\
\int_{\mathbb{R}} \mu^2 \times \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &- \\
2\mu \int_{\mathbb{R}} (x-\mu) \times \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &= \\
\sigma^2 + \mu^2 - 2\mu \times 0 &= \sigma^2 + \mu^2\,. \qquad (1.10)
\end{aligned}$$

1.9.2.2 Entropies

L'entropie (en bits) d'une gaussienne est :

$$\begin{aligned}
-\int_{\mathbb{R}} \phi_{\mu,\sigma^2}(x) \log_2 \phi_{\mu,\sigma^2}(x)\,\mathrm{d}x &= \\
-\frac{1}{\ln 2}\int_{\mathbb{R}} \phi_{\mu,\sigma^2}(x) \ln \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \mathrm{d}x &= \text{// Commentaire : } \\
&\text{// } \log_2(x) = \tfrac{\ln(x)}{\ln(2)} \\
-\frac{1}{\ln 2}\ln \frac{1}{\sqrt{2\pi\sigma^2}} \times 1 - \frac{1}{\ln 2}\int_{\mathbb{R}} \phi_{\mu,\sigma^2}(x) \ln \exp -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \mathrm{d}x &= \text{// On applique l'Eqn. (1.8)} \\
\frac{1}{2\ln 2}\ln(2\pi\sigma^2) - \frac{1}{\ln 2}\frac{-1}{2\sigma^2} \times \sigma^2 &= \text{// On applique l'Eqn. (1.10)} \\
\frac{1}{2\ln 2}\left(\ln(2\pi\sigma^2) + 1\right) &= \text{// Se souvenir que } 1 = \ln e \\
\frac{1}{2\ln 2}\ln(2\pi e\sigma^2) &= \frac{1}{2}\log_2(2\pi e\sigma^2)\,. \qquad (1.11)
\end{aligned}$$

De manière plus générale, l'entropie croisée de deux gaussiennes vaut :

$$
\begin{aligned}
-\int_{\mathbb{R}} \phi_{\mu_1,\sigma_1^2}(x) \log_2 \phi_{\mu_2,\sigma_2^2}(x)\,\mathrm{d}x &= \\
-\frac{1}{\ln 2}\ln\frac{1}{\sqrt{2\pi\sigma_2^2}} \times 1 + \frac{1}{\ln 2}\cdot\frac{1}{2\sigma_2^2}\int_{\mathbb{R}} \phi_{\mu_1,\sigma_1^2}(x)\,(x-\mu_2)^2\,\mathrm{d}x &= \text{// On applique l'Eqn. (1.8)} \\
-\frac{1}{\ln 2}\ln\frac{1}{\sqrt{2\pi\sigma_2^2}} + \frac{1}{\ln 2}\cdot\frac{1}{2\sigma_2^2}\int_{\mathbb{R}} \phi_{\mu_1-\mu_2,\sigma_1^2}(x) \times x^2\,\mathrm{d}x &= \text{// Changement de variable} \\
\frac{1}{2\ln 2}\ln(2\pi\sigma_2^2) + \frac{1}{\ln 2}\frac{\sigma_1^2+(\mu_1-\mu_2)^2}{2\sigma_2^2} &= \text{// On applique l'Eqn. (1.10)} \\
\frac{1}{2\ln 2}\left(\ln(2\pi\sigma_2^2) + \frac{\sigma_1^2+(\mu_1-\mu_2)^2}{\sigma_2^2}\right). & \qquad (1.12)
\end{aligned}
$$

Et donc, on dérive l'expression de la divergence de Kullback-Leibler, comme la différence entre Eqn. (1.12) et (1.11).

1.9.3 Analyse de variance avec bruit additif gaussien

1.9.3.1 Variances inter- et intra-classes avec bruit additif gaussien

Soit X une variable aléatoire discrète et $N \sim \mathcal{N}(0,\sigma^2)$ un bruit blanc gaussien centré. La variable aléatoire $X+N$ suit désormais une loi que l'on appelle « mixture de gaussiennes ».

Notre objectif est d'étudier la propriété 2 d'analyse de variance à la somme $X+N$ de deux réels, conditionnellement à Y.

Tout d'abord, il faut remarquer que la distribution $\mathsf{P}[X+N \mid Y=y]$ est le produit de convolution de la PMF $\mathsf{P}[X \mid Y=y]$ par la gaussienne $N \sim \mathcal{N}(0,\sigma^2)$. Effectivement, $\forall z \in \mathbb{R}$,

$$
\begin{aligned}
\mathsf{P}[X+N=z \mid Y=y] &= \int_{\mathbb{R}} \mathsf{P}[X+N=z \mid Y=y \wedge N=n]\cdot\mathsf{P}[N=n]\,\mathrm{d}n \\
&= \int_{-\infty}^{+\infty} \mathsf{P}[X=z-n \mid Y=y]\cdot\phi_{0,\sigma_2}(n)\,\mathrm{d}n \\
&= \int_{+\infty}^{-\infty} \mathsf{P}[X=x \mid Y=y]\cdot\phi_{0,\sigma_2}(z-x)\times(-1)\,\mathrm{d}x \quad \text{// } x \doteq z-n \\
&= \int_{-\infty}^{+\infty} \mathsf{P}[X=x \mid Y=y]\cdot\phi_{0,\sigma_2}(z-x)\,\mathrm{d}x \quad \text{// } X \text{ est discrète} \\
&= \sum_{x\in\mathcal{X}} \mathsf{P}[X=x \mid Y=y]\cdot\phi_{x,\sigma_2}(z)\,.
\end{aligned}
$$

Ensuite, on calcule d'une part :

$$\begin{aligned}
\mathsf{E}[X+N \mid Y=y] &= \int_{\mathbb{R}} z \times \sum_{x\in\mathcal{X}} \mathsf{P}[X=x \mid Y=y]\cdot \phi_{x,\sigma_2}(z)\,\mathrm{d}z \\
&= \sum_{x\in\mathcal{X}} \mathsf{P}[X=x \mid Y=y]\cdot \int_{\mathbb{R}} z \times \phi_{x,\sigma_2}(z)\,\mathrm{d}z \quad \text{// } \textit{Cf.} \text{ Eqn. (1.9)} \\
&= \sum_{x\in\mathcal{X}} \mathsf{P}[X=x \mid Y=y]\cdot x = \mathsf{E}[X \mid Y=y] \qquad (1.13)
\end{aligned}$$

et d'autre part :

$$\begin{aligned}
\mathsf{E}[(X+N)^2 \mid Y=y] &= \int_{\mathbb{R}} z^2 \times \sum_{x\in\mathcal{X}} \mathsf{P}[X=x \mid Y=y]\cdot \phi_{x,\sigma_2}(z)\,\mathrm{d}z \\
&= \sum_{x\in\mathcal{X}} \mathsf{P}[X=x \mid Y=y]\cdot \int_{\mathbb{R}} z^2 \times \phi_{x,\sigma_2}(z)\,\mathrm{d}z \quad \text{// } \textit{Cf.} \text{ Eqn. (1.10)} \\
&= \sum_{x\in\mathcal{X}} \mathsf{P}[X=x \mid Y=y]\cdot (x^2+\sigma^2) = \mathsf{E}[X^2 \mid Y=y] + \sigma^2\,. \qquad (1.14)
\end{aligned}$$

Ceci permet de montrer que le bruit n'impacte pas la variance inter-classes :

$$\begin{aligned}
\mathsf{V}[\mathsf{E}[X+N \mid Y]] &= \sum_{y\in\mathcal{Y}} \mathsf{P}[y]\cdot(\mathsf{E}[X+N \mid Y=y])^2 - \left(\sum_{y\in\mathcal{Y}} \mathsf{P}[y]\cdot \mathsf{E}[X+N \mid Y=y]\right)^2 \\
&= \sum_{y\in\mathcal{Y}} \mathsf{P}[y]\cdot(\mathsf{E}[X \mid Y=y])^2 - \left(\sum_{y\in\mathcal{Y}} \mathsf{P}[y]\cdot \mathsf{E}[X \mid Y=y]\right)^2 \quad \text{// } \textit{Cf.} \text{ Eqn. (1.13)} \\
&= \mathsf{V}[\mathsf{E}[X \mid Y]]\,, \qquad (1.15)
\end{aligned}$$

mais ajoute sa variance propre à la variance intra-classes :

$$\begin{aligned}
\mathsf{E}[\mathsf{V}[X+N \mid Y]] &= \sum_{y\in\mathcal{Y}} \mathsf{P}[Y=y]\cdot \mathsf{V}[X+N \mid Y=y] \\
&= \sum_{y\in\mathcal{Y}} \mathsf{P}[Y=y]\cdot \big(\mathsf{V}[X \mid Y=y] + \sigma^2\big) \quad \text{// } \textit{Cf.} \text{ Eqn. (1.14)} \\
&= \mathsf{E}[\mathsf{V}[X \mid Y]] + \sum_{y\in\mathcal{Y}} \cancel{\mathsf{P}[y]}\cdot\sigma^2 = \mathsf{E}[\mathsf{V}[X \mid Y]] + \sigma^2\,. \qquad (1.16)
\end{aligned}$$

Ainsi, on obtient la décomposition de la variance suivante :

$$\begin{aligned}
&\underbrace{\mathsf{V}[\mathsf{E}[X+N \mid Y]]}_{\text{variance inter-classes}} + \underbrace{\mathsf{E}[\mathsf{V}[X+N \mid Y]]}_{\text{variance intra-classes}} \\
= \;& \underbrace{\mathsf{V}[\mathsf{E}[X \mid Y]] + \mathsf{E}[\mathsf{V}[X \mid Y]] + \sigma^2 = \mathsf{V}[X] + \mathsf{V}[N]}_{\text{variance totale}}\,. \qquad (1.17)
\end{aligned}$$

1.9.3.2 Théorie de l'information avec bruit additif gaussien

Pour les calculs de théorie de l'information, nous ajoutons une contrainte par rapport à la Sec. 1.9.3.1. En plus de l'« hypothèse gaussienne », qui stipule que le bruit suit une loi normale, nous supposons que l'on peut approximer les distributions (qui sont des mixtures de gaussiennes) par une unique gaussienne. Un critère usuel pour la choix de cette gaussienne est de retenir celle qui minimise la divergence de Kullback-Leibler avec la vraie distribution [389]. Il s'avère qu'il s'agit de la gaussienne de même moyenne et variance que celles de distribution d'origine. On qualifie donc l'approximation de fusion avec préservation des moments d'ordres un et deux (en anglais : *moment-preserving merge*). Nous avons qualifié cette approche d'« approximation gaussienne » (voir notre présentation pionnière dans [278]).

Les distributions X, $X \mid Y = y$ d'une part et N d'autre part étant indépendantes, dans la somme $X + N$, les moyennes et les variances s'ajoutent simplement. Ceci a aussi été démontré dans les Eqn. (1.15) et (1.16). Voilà ainsi les lois intéressantes :

- $X + N \sim \mathcal{N}(\mu_{\text{tot}}, \sigma^2_{\text{tot}} + \sigma^2)$, et
- $\forall y \in \mathcal{Y}, X + N \mid Y = y \sim \mathcal{N}(\mu_y, \sigma^2_y + \sigma^2)$.

Pour rendre les équations compactes, nous avons introduit les notations suivantes :

- $\mu_{\text{tot}} \doteq \mathsf{E}[X]$, $\sigma^2_{\text{tot}} \doteq \mathsf{V}[X]$ et leur pendant conditionnel en $y \in \mathcal{Y}$, à savoir
- $\mu_y \doteq \mathsf{E}[X \mid Y = y]$, $\sigma^2_y \doteq \mathsf{V}[X \mid Y = y]$.

Sous cette hypothèse, le calcul de l'information mutuelle peut être conduit de façon analytique [278], en utilisant le résultat de l'Eqn. (1.11).

$$\begin{aligned}
\mathsf{I}[X+N;Y] &= \mathsf{H}[X+N] - \sum_{y\in\mathcal{Y}} \mathsf{P}[y] \cdot \mathsf{H}[X+N \mid Y=y] \\
&= \frac{1}{2}\log_2 \frac{\cancel{2\pi e}\,(\sigma^2_{\text{tot}} + \sigma^2)}{\prod_{y\in\mathcal{Y}} \cancel{(2\pi e)}^{\mathsf{P}[Y=y]}\,(\sigma^2_y + \sigma^2)^{\mathsf{P}[Y=y]}} \\
&= -\frac{1}{2}\sum_{y\in\mathcal{Y}} \mathsf{P}[y] \cdot \log_2 \frac{\sigma^2_y + \sigma^2}{\sigma^2_{\text{tot}} + \sigma^2} \quad . \qquad (1.18)
\end{aligned}$$

On peut noter que le même résultat aurait pu être obtenu en utilisant la formulation de

TABLE 1.3 – Métriques sur des variables aléatoires gaussiennes.

$\mathsf{E}[X]$	μ_1	*Cf.* Eqn. (1.9)
$\mathsf{V}[X]$	σ_1^2	Soustraction de Eqn. (1.10) et du carré de Eqn. (1.9)
$\mathsf{H}[X]$	$\frac{1}{2}\log_2\left(2\pi e\sigma_1^2\right)$	*Cf.* Eqn. (1.11)
$\mathsf{D}_{\mathsf{KL}}[X \parallel Y]$	$\frac{1}{2\ln 2}\left(\frac{(\mu_1-\mu_2)^2+\sigma_1^2}{\sigma_2^2} - 1 - \ln\frac{\sigma_1^2}{\sigma_2^2}\right)$	Soustraction de Eqn. (1.12) et de Eqn. (1.11)

l'information mutuelle comme une espérance d'une divergence de Kullback-Leibler, *i.e.*

$$
\begin{aligned}
\mathsf{I}[X+N;Y] &= \mathsf{E}[\mathsf{D}_{\mathsf{KL}}[X+N \mid Y \parallel X+N]] \quad // \textit{ Cf. } \text{ligne 3 du Tab. 1.3} \\
&= \sum_{y\in\mathcal{Y}} \frac{1}{2\ln 2}\mathsf{P}[y]\cdot\left(\frac{(\mu_y-\mu_{\text{tot}})^2+(\sigma_y^2+\sigma^2)}{(\sigma_{\text{tot}}^2+\sigma^2)} - 1 - \ln\frac{\sigma_y^2+\sigma^2}{\sigma_{\text{tot}}^2+\sigma^2}\right) \\
&= \frac{1}{2\ln 2}\frac{1}{\sigma_{\text{tot}}^2+\sigma^2}\left(\underbrace{\sum_{y\in\mathcal{Y}}\mathsf{P}[y]\cdot(\mu_y-\mu_{\text{tot}})^2}_{=\mathsf{V}[\mathsf{E}[X+N|Y]]} + \underbrace{\sum_{y\in\mathcal{Y}}\mathsf{P}[y]\cdot\left(\sigma_y^2+\sigma^2\right)}_{=\mathsf{E}[\mathsf{V}[X+N|Y]]}\right) \\
&+ \frac{1}{2\ln 2}\left(-1-\sum_{y\in\mathcal{Y}}\mathsf{P}[y]\cdot\ln\frac{\sigma_y^2+\sigma^2}{\sigma_{\text{tot}}^2+\sigma^2}\right) \\
&= \cancel{\frac{1}{2\ln 2}\frac{1}{\sigma_{\text{tot}}^2+\sigma^2}\left(\sigma_{\text{tot}}^2+\sigma^2\right)} - \cancel{\frac{1}{2\ln 2}} - \frac{1}{2\ln 2}\sum_{y\in\mathcal{Y}}\mathsf{P}[y]\cdot\ln\frac{\sigma_y^2+\sigma^2}{\sigma_{\text{tot}}^2+\sigma^2} \quad ,
\end{aligned}
$$

expression qui est bien identique à l'Eqn. (1.18).

1.9.3.3 Conclusion

Soient X et Y deux variables aléatoires gaussiennes, $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ et $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$. Alors, les deux premiers moments et les grandeurs de la théorie de l'information (en bits) sont résumés dans Tab. 1.3.

Chapitre 2

Curriculum vitæ et publications

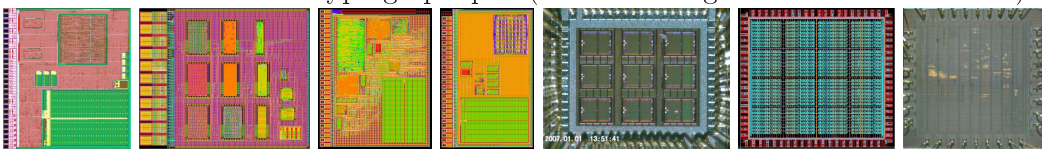
2.1 État civil

Sylvain Guilley, né le 23 novembre 1977 à Strasbourg. Ingénieur en chef des mines. PACSé à Charlotte Baratin, un enfant (Enguerrand).

2.2 Expérience professionnelle

2008– Maître de conférences à TELECOM-ParisTech. Activités de recherche, enseignement et valorisation en électronique de confiance.

2002–2008 Chargé d'enseignement et de recherches à l'ENST. Membre élu de la commission de la recherche de l'ENST. Création d'un laboratoire spécialisé dans l'évaluation de systèmes électroniques. Mise en place des attaques SPA, DPA, template, MIA, EMA et DFA. Responsable du projet structurant GET "trusted computing platform". Fabrication de 7 ASICs cryptographiques (4 en technologie 130 nm et 3 en 65 nm) :



2001 Concepteur électronique à STMicroelectronics, San Diego, CA, USA (stage : 6 mois). Conception et réalisation d'un modem UMTS-FDD pour le standard 3GPP.

2000 Chercheur au centre IBM T. J. Watson, Yorktown, NY, USA. Développement d'un procédé de photo-lithographic à 153 nm (stage : 5 mois).

1999 Aide humanitaire dans l'organisation CISED, Bolivie. Création d'un système d'eau potable dans une communauté Quechua de l'Altiplano (stage : 1 mois).

1997 Officier en charge des opérations, dans la frégate lance-missile Duquesne D603. Exercices dans le bassin méditerranéen et l'océan Atlantique (service militaire : 1 an).

1996 Stagiaire dans le laboratoire de chimie supra-moléculaire du Professeur Jean-Marie Lehn, Strasbourg. Synthèse d'un hélicate (stage : 1 mois).

2.3 Affiliations, prix et distinctions

- Membre des sociétés savantes IEEE et IACR, et membre sénior du club CryptArchi.
- « *Best paper award* » pour l'article "Leakage Squeezing Countermeaure Against High-Order Attacks" présenté au cinquième « *international Workshop in Information Security and Practice (WISTP 2011)* ».
- Finaliste du prix ASTI de thèse 2009.
- Médaille de bronze de la défense nationale, agrafe « bâtiments de combat » (1998).
- Brevet militaire de parachutisme Français, n° 633966, délivré le 29/07/1998 par le Colonel Leroy, commandant l'ETAP (École des Troupes Aéroportées de Pau).

2.4 Formation

2002–2007 Thèse de doctorat, intitulée "Contremesures géométriques aux attaques sur les canaux cachés", dirigée par Renaud Pacalet et soutenue le 10 janvier 2007. Mention très honorable avec les félicitations du jury.

2001–2002 Diplôme d'Études Approfondies (DEA) en physique quantique (Paris-6 / ENS, Laboratoire Kastler-Brossel). Mention bien.

2000–2002 Études d'ingénieur à l'ENST (maintenant TELECOM-ParisTech).

1997–2000 Études d'ingénieur à l'école polytechnique.

2.5 Encadrement scientifique

2.5.1 Jury de thèses de doctorat

1. François-Xavier Aranda (27 septembre 2012), *examinateur*
   → Sujet : « M.A.R.I.S.E. — Méthode Automatisée de Rétro-Ingénierie sur Système Embarqué ».
   → Poste depuis la thèse : « Thales CEACI (CESTI « matériel » de Toulouse), France ».
2. Maxime Nassar (9 mars 2012), *co-directeur*
   → Sujet : « Low-cost Countermeasures against Physical Attacks on Symmetrical and Asymmetrical Cryptography implemented on Altera FPGAs ».
   → Poste depuis la thèse : « BULL Trustway, France ».
3. Olivier Meynard (18 janvier 2012), *co-directeur*
   → Sujet : « Caractérisation et utilisation du rayonnement électromagnétique pour l'attaque de composants cryptographiques ».
   → Poste depuis la thèse : « Ministère de la défense, France ».
4. Shivam Bhasin (14 décembre 2011), *examinateur*
   → Sujet : « Contre-mesures au niveau logique pour sécuriser les architectures de crypto-processeurs dans les FPGA ».
   → Poste depuis la thèse : « Post-doctorat, TELECOM-ParisTech, France ».

5. Aziz Elaabid (7 décembre 2011), *co-encadrant*
   → Sujet : « Attaques par canaux cachés : expérimentations avancées sur les attaques templates ».
   → Poste depuis la thèse : « ATER, Université Paris 8, France ».
6. Youssef Souissi (6 décembre 2011), *co-directeur*
   → Sujet : « Méthodes optimisant l'analyse des crypto-processeurs sur les canaux cachés ».
   → Poste depuis la thèse : « Post-doctorat, TELECOM-ParisTech, France ».
7. Nidhal Selmane (13 décembre 2010), *co-directeur*
   → Sujet : « Global and local Fault attacks on AES cryptoprocessor : Implementation and Countermeasures (online) ».
   → Postes : *après la thèse* « Nétheos, France », *actuel* « BULL, France ».
8. Laurent Sauvage (3 septembre 2010), *directeur*
   → Sujet : « Cartographie électromagnétique pour la cryptanalyse physique (online) ».
   → Poste depuis la thèse : « Ingénieur de recherche, TELECOM-ParisTech, France. ».
9. Victor Lomné (7 juillet 2010), *examinateur*
   → Sujet : « Power and Electro-Magnetic Side-Channel Attacks : threats and countermeasures (online) ».
   → Poste depuis la thèse : « ANSSI, France ».
10. Sumanta Chaudhuri (15 mai 2009), *co-directeur*
    → Sujet : « Asynchronous FPGA Architectures for Cryptographic Applications (online) ».
    → Postes : *après la thèse* « Post-doctorat, IEF, France », *actuel* « Imperial College, London, UK ».

2.5.2 Encadrements de doctorants

J'encadre actuellement :

1. Pablo Rauzy (2012 –)
2. Annelie Heuser (2012 –)
3. Houssem Maghrebi (2009 –)

Les doctorants que j'ai encadrés par le passé sont listés dans la section précédente (§ 2.5.1).

2.5.3 Encadrements de stages de M2 *et anciens DEAs* (liste partielle)

- Qi Zhou : "Évaluation du masquage aléatoire comme contre-mesure aux attaques de canaux auxiliaires de FPGA implémentant de la cryptographie", LIP6 (2008).
- Zhiguo Song : "Caractérisation du profil de consommation d'un micro-processeur", LIP6 (2007).

- Korinna Lenz (*Diplomarbeit*) : "Differential Power Analysis Attack Against a Software Implementation of DES", Deutsche Telekom AG & University of Applied Sciences Leipzig, Germany (2006).
- Maxence Batiste : "Pilotage d'une expérience de cryptographie quantique par une carte FPGA", (2006).
- Viet Hung Pham : "Réalisation d'un logiciel de téléphonie sur IP sécurisé quantiquement", (2006).
- Saya de León Seta : "Spécification et conception d'un FPGA asynchrone", ENSTA (2005).

2.6 Enseignement

- **Cours à TELECOM-ParisTech** :
  - Création de l'unité d'enseignement ELECINF359 (Sécurité des Systèmes Embarqués) :
    - Cinq cours magistraux sur les algorithmes AES & DES, les attaques DPA & DFA, et la rétroconception (RE) ; cours donnés annuellement depuis 2005.
  - Tronc commun ENI (Électronique Numérique Intégrée) :
    - Encadrement d'un groupe d'élèves (une petite classe) en travaux dirigés et en travaux dirigés ; une vingtaine de tranches horaires données en 2004, 2005 et 2006.
  - Formation continue :
    - Sécurité des Systèmes Embarqués : une journée de cours, donnée en 2006 (21 mai), 2007 (27 septembre), 2008 (9 juin), 2009 (25 mai), 2010 (17 mai et 16 novembre) et 2011 (10 mai, 15 novembre et 15 décembre).
    - Conception ASIC : une journée de cours, donnée en 2006.
    - Conception Verilog : une journée de cours, donnée en 2006.
  - **Cours à Supélec Rennes** :
    - Trois heures et demie de cours sur les contremesures aux attaques physiques de composants dans le cadre du module de formation continue ER10 ("*Confidentialité et sécurité des informations dans les systèmes électroniques numériques intégrés*", juin 2012).
  - **Cours à l'École Nationale Supérieure des Mines de Saint-Étienne** :
    - Master international SISA (*Security of Integrated Systems & Applications*), trois jours de cours et de travaux dirigés (tous deux en langue anglaise), donnés en 2008.
  - **Cours à l'Université Paris VI** :
    - Master LIP6/ASIM. Le langage Verilog : deux demi-journées, en 2005.

2.7 Implication scientifique

2.7.1 Présidences de comités de programme de conférences

- **SPACE 2013**, avec Debdeep Mukhopadhyay et Benedikt Gierlichs,
- **FDTC 2011**, avec Junko Takahashi,
- **JNRDM 2005**.

2.7.2 Service dans des comités de programme de conférences

1. **FPS 2013** (International Symposium On Fundations & Practice of Security) ;
   Sponsor : Springer LNCS.
2. **HOST 2008–2013** (Hardware-Oriented Security and Trust) ;
   Sponsors : TTTC, IEEE Computer Society, IEEE Security and Privacy.
3. **COSADE 2013** (International workshop on Constructive Side-Channel Analysis and Secure Design) ;
   Sponsor : IACR (International Association for Cryptologic Research).
4. **ICISTM 2012–2013** (International Conference on Information Systems, Technology & Management) ;
   Sponsor : Grenoble école de management, MDI Gurgaon, University of Florida ; actes Springer CCIS.
5. **ReConFig 2008–2012** (International Conference on ReConFigurable Computing and FPGAs) ;
   Sponsor : the IEEE Computer Society.
6. **FDTC 2009–2012** (Fault Diagnosis and Tolerance in Cryptography) ;
   Sponsor : the IEEE Computer Society.
7. **CHES 2012** (Workshop on Cryptographic Hardware and Embedded Systems) ;
   Sponsor : IACR (International Association for Cryptologic Research).
8. **RAW 2012** (Reconfigurable Architectures Workshop) ;
   Sponsor : the IEEE Computer Society.
9. **HASP 2012** (Workshop on Hardware and Architectural Support for Security and Privacy) ;
   Sponsor : IEEE.
10. **PHISIC 2013** and **PHISIC 2011** (Practical Hardware Innovations in Security Implementation & Characterisation) ;
    Sponsor : le pôle de compétitivité SCS (Solutions Communicantes Sécurisées, région PACA).
11. **CARDIS 2011** (Smart Card Research and Advanced Application Conference) ;
    Sponsor : Springer LNCS.
12. **DATE 2008–2010** (Design Automation and Test in Europe) ;
    Sponsors : the European Design and Automation Association, the EDA Consortium, the IEEE Council on EDA, ECSI, ACM-SIGDA et RAS.

2.7.3 Évaluation de soumissions de pairs (*i.e.* "*sub-reviews*")

1. le journal **COMPJ** (The Computer Journal), Oxford University Press, The British Computer Society,
2. le journal **Applied Mathematics & Information Sciences**,
3. le journal **TCAS1** (IEEE Transactions on Circuits and Systems I),
4. le journal **INS** (Information Sciences, Elsevier),
5. le journal **TC** (IEEE Transactions on Computers),
6. le journal **JSS** (Journal of Systems and Software, Elsevier),
7. le journal **TIFS** (IEEE Transactions on Information Forensics & Security),
8. le journal **TSI** (Technique et Science Informatiques, Hermes Science),
9. le journal **IET-CDT** (IET Computers & Digital Techniques),
10. le journal **TVLSI** (IEEE Transactions on Very Large Scale Integration (VLSI) Systems),
11. le journal **TECS** (ACM Transactions on Embedded Computing Systems),
12. le journal **TRETS** (ACM Transactions on Reconfigurable Technology and Systems),
13. le journal **PIEEE** (Proceedings of the IEEE),
14. le journal **JCEN** (Journal of Cryptographic Engineering, Springer),
15. le journal **Integration** (the VLSI Journal, Elsevier),
16. la conférence **COSADE 2012** (Third International Workshop on Constructive Side-Channel Analysis and Secure Design),
17. la conférence **ASIACRYPT 2010** (Annual International Conference on the Theory and Application of Cryptology and Information Security),
18. la conférence **ICT 2010** (Information and Coding Theory),
19. la conférence **ICECS 2009** (International Conference on Electronics, Circuits, and Systems),
20. la conférence **ACNS 2009** (International Conference on Applied Cryptography and Network Security),
21. la conférence **InsCrypt 2008** (International Conferences on Information Security and Cryptology),
22. la conférence **SAC 2008** (Selected Areas in Cryptography),
23. la conférence **CARDIS 2008** (Smart Card Research and Advanced Application Conference),
24. les conférences CHES (Cryptographic Hardware and Embedded Systems – IACR), **2007**, **2008**, **2009** et **2010**,
25. la conférence **FSE 2007** (Fast Software Encryption) et
26. la session "hardware security" de la conférence **DATE** (Design, Automation an Test in Europe).

2.7.4 Présidences de sessions en conférences

- **NIAT 2011** ;
- **Cryptarchi 2011** ;
- **WISTP 2011** ;
- **FDTC 2010, 2012** ;
- **COSADE 2010**.

2.7.5 Expertises

J'ai été sollicité comme expert pour l'agence nationale de la recherche (ANR) :

- Programme "Ingénierie Numérique et Sécurité" (INS), 2012 ;
- Programme "Réseaux du Futur et Services" (VERSO), 2009 ;
- Programme "Architectures du futur" (ARFU), 2007.

En 2012, j'ai notamment expertisé à mi-parcours les projets VERSO 2009 suivants : ARSSO, BEST, ViPEER, KIDPOCKET, METAVEST, ECLIPSES et THID.

2.7.6 Concours international de DPA, le « *DPA contest* »

Ce concours a été lancé en 2008 avec Laurent Sauvage et Florent Flament. Depuis le début, je participe à son organisation annuelle. Dans la première édition, il s'agissait d'évaluer les attaques les plus rapides. Le nombre impressionnant de participants (plus d'une vingtaine) a justifié la tenue d'une session plénière lors de CHES 2009. La seconde version du concours a bénéficié de l'aide de Guillaume Duc, qui chapeaute désormais l'organisation de toutes les éditions. Cette fois-ci, le concours utilisait différentes métriques pour caractériser les attaques soumises. L'engouement a aussi motivé une session spéciale, cette fois-ci à COSADE 2011. La troisième version a été lancée en 2011 : elle vise à caractériser les meilleures méthodes d'acquisition de signaux compromettants. Un point informel a été donné pendant les *rump sessions* de CHES 2011 et CHES 2012. L'AIST japonaise a contribué à l'organisation. Le lancement d'une quatrième version est prévu pour début 2012 [114].

Les données (traces ou programmes) du DPA contest ont été utilisées par de nombreux chercheurs, aussi bien dans le monde académique que dans celui de la vulgarisation (voir l'article "Attaques par canaux auxiliaires" intitulé « *If it leaks, we can kill it* », de Rémy Daudigny, dans le H.S. #5 de la revue MISC).

2.8 Publications

Un résumé de mes travaux publiés se trouve dans le tableau 2.1. Ceux-ci m'attribuent un H-index de 9 (d'après le service de base de donnée Scopus de l'éditeur Elsevier).

TABLE 2.1 – Résumé des travaux publiés.

Type de publication	**Nombre & Références**
Journaux	13 ...[38, 119, 397, 410, 402, 210, 101, 399, 175, 187, 195, 203, 150]
Brevets	10 .....[179, 181, 178, 94, 177, 98, 183, 93, 100, 105]
Dépôts de logiciels	5 .....[102, 163, 223, 185, 184]
Chapitres de livres	3 .....[104, 176, 95]
Conférences avec actes	81 [108, 36, 325, 66, 285, 56, 58, 82, 273, 284, 394, 427, 324, 182, 334, 423, 282, 42, 332, 109, 424, 425, 393, 310, 278, 209, 351, 197, 280, 199, 314, 80, 396, 431, 37, 315, 3, 215, 281, 206, 400, 118, 426, 429, 39, 311, 331, 171, 401, 33, 97, 274, 411, 43, 395, 235, 186, 29, 242, 205, 208, 204, 172, 75, 173, 412, 72, 73, 76, 30, 70, 99, 222, 196, 476, 117, 71, 191, 190, 194, 193]
Conférences sans actes	33 .[59, 169, 114, 168, 107, 41, 166, 92, 40, 428, 110, 120, 35, 198, 312, 115, 164, 201, 275, 333, 200, 127, 31, 211, 213, 103, 482, 153, 126, 349, 157, 158, 155]
Séminaires informels	12 ...[167, 202, 313, 32, 165, 212, 214, 180, 170, 161, 159, 160]
Rapports	6 .....[276, 74, 77, 221, 2, 156]
Thèses (PhD & MSc)	2 .....[162, 154]

2.9 Vulgarisation

J'ai contribué à faire connaître la discipline au travers de diverses actions de vulgarisation, notamment en écrivant des communiqués de presse, repris par différents médias.

1. **Presse locale de Nara, Kansai, Japon**, suite aux présentations poster de CHES 2011. Voir Fig. 2.1.
2. "Piratage de cartes à puces : DPA Contest", **Data Security Breach**, magazine d'actualité liée au cybercrime, pirates, hackers et sécurité de nos données sur Internet, 1er février 2012. Voir Fig. 2.2.
3. "L'Institut Télécom lance une offensive contre le piratage des cartes à puces", **Le Magazine de la Sécurité Informatique**, Octobre 2011. Voir Fig. 2.3.
4. "Un concours international de cryptologie, organisé par l'Institut Télécom, pour aider à parer aux attaques sur les cartes à puce", **Le Magazine de la Sécurité Informatique**, Septembre 2009. Voir Fig. 2.4.
5. "SecLib renforce la sécurité des circuits cryptographiques", **l'Atelier BNP Paribas**, décembre 2008. Voir Fig. 2.5.
6. "L'Institut TELECOM dévoile ses circuits cryptographiques les plus robustes", **communiqué de presse**, 11/12/2008. Voir Fig. 2.6.
7. "Création de masters et de masters professionnels", **01Informatique**, août 2006. Voir Fig. 2.7.
8. "Bouchons anti-fuites pour circuits électroniques", encart dans le numéro "Spécial 40 ans" de **01Informatique**, 16 juin 2006, numéro 1864, page 104. Voir Fig. 2.8.
9. "Point de vue sur la gestion des droits numériques", **TELECOMEDIA** numéro 14, mai-août 2003. Voir Fig. 2.9.

2.10 Projets de recherche collaboratifs

J'ai contribué à la rédaction des annexes scientifiques des projets de recherche collaborative suivants :

- **PISCO**, *Plateforme d'Intégration de Services de Confiance*, projet FUI (AAP n° 14), avec BULL S.A.S., Bertin Technologies, CASSIDIAN S.A.S., CS Communication et Systèmes, S.A.R.L. Serpikom, Cryptolog International, SafeRiver, Oppida, CEA LIST, TELECOM-ParisTech et INRIA.
- **HOMER**, *Hardware TrOjans, MEnaces et Robustesses des circuits intEgrés*, projet FUI (AAP n° 14), avec CASSIDIAN CyberSecurity, Gemalto, Secure-IC, ANSSI, ARMINES, CEA-LETI, LIRMM, TELECOM-ParisTech.
- **PEARL**, *Platform for Embedded Application with high Robustness Level*, projet ITEA-2, avec Cassidian, Bull, Morpho, Secure-IC, Ctech, TWT, Bosch, Ifak, Secure-IC.


奈良新聞 2011.9.30

県独自の国際会議エリア初会合

歴史や物語に高評価

誘致促進に弾み

奈良

FIGURE 2.1 – Presse locale de Nara, Japon, suite aux présentations poster de CHES 2011.

Piratage de cartes à puces : DPA Contest

Written by admin. Posted in Actualités, Carte à puce

Tagged: carte a puce, crack, dpa contest, piratage, securite

Published on février 01, 2012 with No Comments

1 tweet retweet

A l'initiative de l'Institut Télécom, les chercheurs se mobilisent contre le piratage des cartes à puces jusqu'à fin février 2012. Le piratage des cartes à puce est une menace pour les données privées des particuliers et pour la qualité des services des opérateurs (téléphonique, banque, monétique, billettique, accès conditionnels). Pour lutter contre ce fléau, l'Institut Télécom organise depuis trois ans un concours de recherche appelé DPA Contest et destiné aux chercheurs en cyberattaque du monde entier. La démarche repose sur un constat très simple, ce que DataSecurityBreach.fr vous explique depuis longtemps : il est illusoire de chercher à se protéger si l'on n'a pas compris pourquoi on est vulnérable. La troisième édition de ce concours se termine fin février 2012 et ses résultats seront présentés en mai.

Comprendre les attaques et les techniques d'acquisition pour mieux se défendre

Ce concours de cryptologie, organisé par des chercheurs en systèmes électroniques numériques de Télécom ParisTech (Jean-Luc Danger, Guillaume Duc, Sylvain Guilley), vise à étudier les attaques par « *canaux auxiliaires* » contre les systèmes embarqués tels que les cartes à puce. En effet, lorsqu'un circuit électronique fonctionne, l'énergie consommée est observable sur les canaux dits auxiliaires comme l'intensité du courant ou le champ électromagnétique. Cette activité énergétique dépend des calculs et des données manipulées dont l'information sensible qui fuit inévitablement par ce biais.

Lors les deux premières éditions du concours, de nombreuses équipes de chercheurs internationaux, aussi bien académiques qu'industrielles dans le domaine de la sécurité des systèmes embarqués, ont développé et soumis des programmes informatiques analysant ces fuites d'information afin de retrouver un secret (souvent une clé de chiffrement) enfoui au sein du circuit électronique.

Mais l'acquisition de ces signaux est aussi une étape cruciale dans le succès d'une telle attaque. C'est pour cette raison que cette édition 2011-2012, organisée pour la première fois en partenariat avec le National Institute of Advanced Industrial Science and Technology au Japon, est tout particulièrement dédiée aux techniques de métrologie et de traitement des signaux. Ce concours permet dans un premier temps de cataloguer et de mieux comprendre diverses stratégies d'attaque.

Fin du 3e concours en février 2012 et présentation des résultats en mai 2012

La présentation des résultats du concours 2011-2012 aura lieu en mai 2012 an Allemagne lors du workshop international *COSADE* qui réunira des chercheurs de tous les horizons sur le sujet du « *secure design*« .

Vers une étude approfondie et méthodique des contre-mesures

La nouvelle édition du concours DPA Contest, lancée début 2012 pour se terminer début 2013, concerne l'étude de quelques contre-mesures proposées par des académiques ou des industriels pour tenter de se protéger contre ces attaques par canaux auxiliaires, et sélectionnées par un comité d'experts scientifiques. Ces contre-mesures seront testées et comparées de manière équitable, ce qui permettra de faire ressortir les faiblesses ou les forces de chacune.

FIGURE 2.2 – Publication du 1er février 2012, dans http://datasecuritybreach.fr/.

LE MAGAZINE DE LA SÉCURITÉ INFORMATIQUE vendredi 21 octobre 2011 **Connexion** || **Inscription**

MAG SECURS
INFORMATIQUE ■ RÉSEAUX ■ TELECOM ■ INTERNET

Salle de presse

Accueil **News** **Alertes** **Tests** **Communiqués** **Livres Blancs** **Dossiers** **Agenda** **Formation** **Emploi**

PROTECTION DES DONNÉES **GESTION D'IDENTITÉ** **GESTION DES ACCÈS** **SÉCURITÉ PHYSIQUE** **GOUVERNANCE**

Communiqués

Articles courants | **Archives** | **Recherche**

L'Institut Télécom lance une offensive contre le piratage des cartes à puces

Le piratage des cartes à puce est une menace pour les données privées des particuliers et pour la qualité des services des opérateurs (téléphonique, banque, monétique, billetique, accès conditionnels). Pour lutter contre ce fléau, l'Institut Télécom a lancé une grande opération de recherche via un concours public appelé DPA Contest. Organisé par les chercheurs de Télécom ParisTech, c'est la deuxième édition cette année qui est en train de s'achever (fin du concours le 31 octobre).

Pendant 12 mois, des équipes de recherche du monde entier ont été appelées à étudier la sécurité d'un algorithme de chiffrement, l'AES (Advanced Encryption Standard), largement utilisé dans les systèmes embarqués tels que les cartes à puce.

Comprendre les attaques pour mieux se défendre

Ce concours de cryptologie vise à recueillir des codes d'attaques sur des mesures de canaux auxiliaires. Ces mesures sont des signaux faibles, se manifestant physiquement comme des fluctuations sur l'alimentation ou un rayonnement électromagnétique, qui fuient de façon incontrôlable du cœur même du circuit cryptographique.

Pour le besoin du concours, l'équipe de chercheurs de Télécom ParisTech, dirigée par Jean-Luc Danger, a réuni en nombre conséquent (plusieurs millions de mesures) de telles émanations. Le but de ce concours est d'en extraire au plus vite la clé secrète de l'algorithme AES (Advanced Encryption Standard).

L'objectif de cette démarche repose sur un constat très simple : il est illusoire de chercher à se protéger si l'on n'a pas compris pourquoi on est vulnérable. Ainsi, ce concours permet-il dans un premier temps de cataloguer divers stratégies d'attaque. Les participants sont bien entendu invités à détailler leurs exploits.

Vers un diagnostic méthodique de la sécurité des cartes à puces

Mais la réelle spécificité de cette seconde édition du DPA contest réside dans la richesse des "benchmarks". En effet, dans un deuxième temps, les organisateurs analysent dans les moindres détails les forces et faiblesses de chaque attaque. Pour bien comprendre les attaques, différents critères de succès et de performance ont été définis et sont calculés de façon exhaustive. Ces critères permettent également de comparer les attaques entre elles. En plus du classement des attaques entre elles, ces critères permettent donc de rôder une méthodologie de diagnostique sécuritaire.

Effectivement, les outils de caractérisation développés à l'Institut Télécom permettront à l'avenir de tester la qualité des contre-mesures face à ces attaques.

Ce travail a été réalisé de concert par Guillaume Duc et Sylvain Guilley de Télécom ParisTech et l'équipe de cryptologues de l'Université catholique de Louvain.

Une mobilisation internationale aussi bien publique que privée

Plus d'une vingtaine d'attaques ont été soumises depuis le début du concours et les tactiques employées sont très variées, comme en témoigne la vitesse d'exécution des attaques : de 1 à 100 ! Il est notable de remarquer que la sécurité des cartes à puce mobilise aussi bien industriels qu'académiques, et que les contributions sont vraiment internationales. La problématique de la sécurité des composants est donc réellement un

Partenaires Mag-Securs

terrain d'entente pour de multiples collaborations. Et de nombreux projets de recherche approfondissant notamment les parades aux attaques utilisent d'ailleurs les données du DPA Contest comme une plateforme de référence au-delà même du concours.

Fin du concours le 31 octobre et présentation des résultats en février 2011
La présentation des résultats du concours 2010 aura lieu les 24 et 25 février 2011 en Allemagne lors du workshop international COSADE qui réunira chercheurs de tous horizons sur le sujet du « secure design » (http://cosade2011.cased.de/). L'équipe de Télécom ParisTech publiera à cette occasion un document de synthèse du concours et lancera l'édition 2011.

Un pas de plus avec l'aide des partenaires Japonais dès 2011
Pour la 3e édition en 2011, les Japonais, qui étaient déjà présents au concours en 2010, proposeront d'ajouter au volet mathématique habituel un volet « physique » lié à la captation des signaux compromettants.
Les équipes japonaises ont notamment développé une carte d'évaluation des fuites d'information des calculs cryptographiques, nommée SASEBO et d'ores et déjà diffusée auprès des équipes académiques qui participe au DPA Contest 2010 pour leurs expérimentations. L'idée est de permettre aux équipes intéressées de soumettre également des acquisitions de courants ou de rayonnements compromettants.

Le DPA Contest prendra ainsi une nouvelle dimension dans la communauté scientifique.

IMPORTANT : FIN DU CONCOURS le 31 octobre 2010.
Plus d'information sur http://www.dpacontest.org/v2/

Contact presse : Pleon
Célia Casabianca - +33 (0)1 53 32 62 06 – celia.casabianca@pleon.com
Esla Portal - +33 (0)1 53 32 64 66 – elsa.portal@pleon.com
Institut Télécom
Jérôme Vauselle - +33 (0)1 45 81 75 05 – jerome.vauselle@institut-telecom.fr

A propos de l'Institut Télécom www.institut-telecom.fr

L'Institut Télécom est un organisme d'enseignement supérieur et de recherche en sciences et technologies de l'information et de la communication. Il regroupe les grandes écoles Télécom ParisTech, Télécom Bretagne, Télécom SudParis et Télécom Ecole de Management ainsi que deux filiales Télécom Lille1 et Eurecom soit 5550 étudiants, 650 enseignants-chercheurs et 950 doctorants, post-docs et sabbatiques. Acteur européen de référence dans son domaine, l'Institut Télécom constitue depuis 2008 un réseau d'écoles associées : Télécom Saint-Etienne, ENSPS (Strasbourg), ENSEIRB-MATMECA (Bordeaux), Sup'Com Tunis (Tunisie) et l'INP-ENSEEIHT (Toulouse). Tourné vers l'innovation, l'Institut Télécom a été labellisé Institut Carnot avec sa filiale Eurecom en 2006 pour la qualité de sa recherche partenariale et crée plus de 50 start-up par an dans ses incubateurs.

FIGURE 2.3 – Mag. Securs, octobre 2011.

prochains sommaires
abonnements - publicité

Sociétés

Un concours international de cryptologie, organisé par l'Institut Télécom, pour aider à parer aux attaques sur les cartes à puce

août 2009

Depuis plusieurs mois, des chercheurs de Télécom ParisTech ont lancé un concours international d'évaluation de la sécurité des composants cryptographiques embarqués : le DPA Contest. Un événement majeur pour les chercheurs mais aussi les professionnels, notamment de la banque, dans un univers où la culture du secret domine. Les premiers résultats seront restitués du 6 au 9 septembre 2009 lors de la conférence CHES (Workshop on Cryptographic Hardware and Embedded Systems).

Quelle sécurité pour nos cartes bleues et nos téléphones ?

Les composants de cryptographie embarqués sont très présents dans notre quotidien car on les retrouve dans les puces de carte bleue ou de téléphone pour protéger les données et les communications. Pour autant, ils sont vulnérables face à des attaques via le courant électrique consommé ou les ondes électromagnétiques. Différentes parades théoriques ont été publiées du côté des chercheurs, mais leur réelle efficacité reste délicate à jauger. Du côté des industriels, chacun a sa recette secrète mais les attaques ont au moins une longueur d'avance sur les contre-mesures. Il devient dès lors primordial de disposer d'une méthodologie standard d'évaluation sécuritaire.

Déclois onner en mettant en commun les méthodes et les résultats

Pour Jean-Luc Danger et Sylvain Guilley, chercheurs à Télécom ParisTech, *« dans un univers où la culture du secret domine, il est difficile de confronter des approches concurrentes. Ce concours vise à remédier au cloisonnement actuel en promouvant l'examen critique des recettes propriétaires par un accès complet à leurs spécifications, un peu comme dans le domaine du logiciel libre »*. La démarche du DPA Contest est donc imprégnée du désir d'établir davantage de transparence dans le domaine émergeant de la sécurité des implémentations. Le passage d'une standardisation des primitives cryptographiques à celle de leurs implémentations est à la clé ! Et intéressera autant les banques que les domaines de la sécurité d'Etat...

Pour atteindre cet objectif, le site web http://www.dpacontest.org/ propose des mesures de canaux auxiliaires accessibles publiquement. Elles font office de référence pour évaluer la force d'une attaque. Dans le moyen terme, elle pourrait servir de support standardisé. Plus de 100 000 mesures de référence peuvent être ainsi téléchargées depuis un serveur de base de données. Toute personne ou établissement désireux de profiter de cette dynamique est invité à participer.

Pour Francis Jutand, directeur scientifique de l'Institut Télécom « c'est une façon pour l'Institut Télécom de faire profiter la communauté internationale de son expérience cumulée dans le domaine de la sécurité des composants cryptographiques, et plus particulièrement pour les laboratoires de Télécom ParisTech d'affirmer leur expertise dans la conception de systèmes de sécurité ». L'Institut Télécom joue ainsi pleinement son rôle académique en suscitant un travail ouvert (basé sur l'accès aux codes sources) et collaboratif (chacun peut contribuer librement en déposant en ligne son algorithme d'attaque).

Chercheurs et spécialistes se retrouvent à Lausanne pour les résultats du concours 2009

Depuis le lancement du concours en août 2008, 3700 personnes ont consulté le web et/ou téléchargé les données de recherche depuis 43 pays, comme la France, le Japon, l'Allemagne, les Pays-Bas, les Etats-Unis, la Belgique, la Chine, la Corée du Sud... A ce jour, 30 soumissions originales ont été reçues. Les participants au concours sont autant des laboratoires publics que privés.

Les résultats du DPA Contest seront annoncés lors de la session plénière de la conférence phare du domaine (CHES) qui aura lieu à Lausanne du 6 au 9 septembre et à laquelle plus de 250 personnes, chercheurs et spécialistes de la cryptographie, participeront. Sylvain Guilley présentera le

vainqueur du concours avant de lancer l'édition 2010. Plus d'information sur la conférence : http://www.chesworkshop.org/

Telecom Paris Tech
août 2009

< article précédent article suivant >

0 commentaire(s) ajouter un commentaire ou poser une question

FIGURE 2.4 – Mag. Securs, septembre 2009.

SecLib renforce la sécurité des circuits cryptographiques

Par L'Atelier BNP Paribas - Paris - 26/12/2008 - 15h05

Développée par l'Institut Telecom, cette technologie rend les cartes à puce et autres terminaux électroniques portables aussi immunes que possible aux assauts des cybercriminels.

Retrait d'espèces, passage à un portique de transports en commun ou identification par le biais de sa carte Vitale sont autant d'actes quotidiens qui font intervenir des circuits cryptographiques (clés mathématiques) nécessaires à la sécurisation des données. Problème : ces circuits doivent être protégés contre les attaques "physiques" qui tirent parti de l'information véhiculée par le courant électrique et le rayonnement électromagnétique qu'il émet. L'institut Télécom a donc mis au point un système de contre-mesure visant

Sécurisation des données

Selon l'institut Télécom, l'attaque, en nombre de mesures à effectuer, pour éventuellement mettre à mal SecLib est en effet d'au moins trois cent cinquante fois supérieure. Le système dit de "parades à l'introspection maligne" mis en œuvre par SecLib combine une logique de calcul à activité constante avec une complète symétrisation des chemins de données. Ses inventeurs expliquent par ailleurs qu'un soin particulier a été apporté à l'équilibrage de l'apport d'énergie et à l'isolation électrique des signaux contre la diaphonie. Ce qui permet de fournir la meilleure résistance possible contre les attaques de l'état de l'art. Pour mettre au point ce système, l'institut Télécom s'est associé pendant cinq ans avec le fabricant de circuits intégrés Franco-Italien STMicroelectronics.

Des algorithmes mathématiquement robustes

C'est ainsi que la logique SecLib a pu être validée sous diverses formes dans les circuits durcis de la famille "SecMat" (Securité du Matériel) ASIC en technologie 130 nanomètres. Pour mémoire, même si les algorithmes utilisés sont mathématiquement robustes, il est possible de compromettre la sécurité d'un circuit cryptographique en espionnant son fonctionnement interne à l'aide de matériel de laboratoire (sondes de courant, antennes, oscilloscopes, etc.). A noter enfin : quatre chercheurs de Télécom ParisTech se sont rassemblés pour créer Secure-IC. Cette spin-off a pour objectif de fournir des services aux grandes entreprises de la Défense afin de protéger, au-delà des cartes à puces, tous les circuits à "haute performance" contre des menaces similaires.

URL source: http://atelierlabs.fr/fr/articles/seclib-renforce-securite-circuits-cryptographiques

FIGURE 2.5 – Annonce sur la logique “SecLib”, par l’Atelier BNP Paribas.

INSTITUT TELECOM

Paris, le 11 décembre 2008

COMMUNIQUE DE PRESSE

> L'Institut TELECOM dévoile ses circuits cryptographiques les plus robustes

Le retrait d'espèces à un distributeur de billets, le passage à un portique de transports en commun avec un badge sans contact, l'identification chez un médecin par le biais d'une carte Vitale, ou encore le téléchargement d'une chanson sur un baladeur numérique sont autant d'actes quotidiens qui font intervenir des circuits cryptographiques (clés mathématiques) nécessaires à la sécurisation des données. Mais ces circuits doivent être protégés contre les attaques « physiques » qui tirent partie de l'information véhiculée par le courant électrique et le rayonnement électromagnétique qu'il émet.

Protéger les circuits cryptographiques grâce à des logiques électroniques sécurisées

Qu'ils revêtent la forme d'une carte à puce ou d'un terminal électronique portable, les circuits cryptographiques permettent d'engager des transactions en toute confiance. Pourtant, de nombreux acteurs malveillants sont en permanence tentés d'abuser de ces facilités dans l'optique de se faire passer de façon illégitime pour quelqu'un d'autre ou simplement pour contourner une mesure technique de protection. Même si les algorithmes utilisés sont mathématiquement robustes, il est possible de compromettre la sécurité d'un circuit cryptographique en espionnant son fonctionnement interne à l'aide de matériel de laboratoire (sondes de courant, antennes, oscilloscopes, etc.).

Les chercheurs de TELECOM ParisTech développent précisément des contre-mesures visant à rendre les circuits cryptographiques aussi immunes que possible à ce type d'assauts de cybercriminels. Des styles de logique électronique, dont la consommation électrique est rendue aussi équilibrée que possible, ont ainsi été conçus et testés pour garantir la sécurité des systèmes.

La meilleure résistance aux attaques de l'état de l'art

Les cartes à puces actuellement disponibles sur la marché proposent de nombreux chausse-trappes et autres mécanismes de prévention d'attaques. Parmi toutes les solutions possibles pour lutter contre les attaques en observation de la carte, TELECOM ParisTech a exploré et perfectionné celle qui garantit une consommation constante ; dans ce créneau, les contre-mesures inventées à l'Institut TELECOM résistent le mieux aux attaques de l'état de l'art. Les parades à l'introspection maligne mises en œuvre, rassemblées sous le terme de SecLib (Secured Library), combinent une logique de calcul à activité constante avec une complète symétrisation des chemins de données et un soin extrême apporté à l'équilibrage de l'apport d'énergie et à l'isolation électrique des signaux contre la diaphonie.
L'évaluation sécuritaire la plus poussée montre que la logique SecLib est au moins dix fois plus solide que la logique sécurisée WDDL, très étudiée dans le monde académique, et que la puissance d'une attaque, en nombre de mesures à effectuer, pour (éventuellement) mettre à mal SecLib est d'au moins 350 fois supérieure à celle d'un style de logique de référence.
[Référence bibliographique : http://doi.ieeecomputersociety.org/10.1109/TC.2008.109]

Une collaboration de plusieurs années avec STMicroelectronics

Fruit de 5 années de collaboration avec le fabricant de circuits intégrés Franco-Italien STMicroelectronics, la logique SecLib a été validée sous diverses formes dans les circuits durcis de la famille "SecMat" (Securité du Matériel), ASIC en technologie 130 nanomètres. Ces « cartes à puces académiques » ont permis de mettre en œuvre nombre d'attaques sur des prototypes matériels et d'évaluer la robustesse des contre-mesures.

De la recherche à la création d'entreprise innovante

Profitant de leur expertise dans le domaine de la sécurité, 4 chercheurs de TELECOM ParisTech se sont rassemblés pour créer Secure-IC (http://www.secure-ic.com/). Cette *spin-off* a pour objectif de fournir des services aux grandes entreprises de la Défense afin de protéger, au-delà des cartes à puces, les circuits à "haute performance", tels que les FPGA, contre des menaces similaires.

Contact presse : Agence Point Virgule
Chrystel Libert – +33 (0)1 73 79 50 63 – clibert@pointvirgule.com
Solenn Morgon – +33 (0)1 73 79 50 70 – smorgon@pointvirgule.com
Institut TELECOM
Jérôme Vauselle - +33 (0)1 45 81 75 05 – jerome.vauselle@institut-telecom.fr

A propos de l'Institut TELECOM www.institut-telecom.fr
L'Institut TELECOM est un organisme d'enseignement supérieur et de recherche en sciences et technologies de l'information et de la communication (STIC). Il regroupe les grandes écoles TELECOM ParisTech, TELECOM Bretagne, TELECOM SudParis et TELECOM Ecole de Management ainsi que deux filiales TELECOM Lille1 et EURECOM soit 5000 étudiants, 600 enseignants-chercheurs et 800 doctorants, post-docs et sabbatiques. Depuis mai 2008, l'Institut TELECOM, acteur européen de référence en STIC, compte également deux écoles associées : TELECOM Saint-Etienne et l'ENSPS.
TELECOM ParisTech : première grande école française d'ingénieurs dans le domaine des sciences et des technologies de l'information et de la communication (STIC), TELECOM ParisTech forme les ingénieurs de la société de l'information. L'enseignement prépare les étudiants à devenir acteurs du domaine des STIC, aujourd'hui omniprésent et facteur de croissance rapide de l'économie. L'école, membre fondateur de ParisTech, aux côtés de onze des plus prestigieuses grandes écoles françaises, est aussi un centre de recherche reconnu internationalement et héberge deux incubateurs.

FIGURE 2.6 – Communiqué de presse de l'Institut Télécom au sujet de SecMat V3.

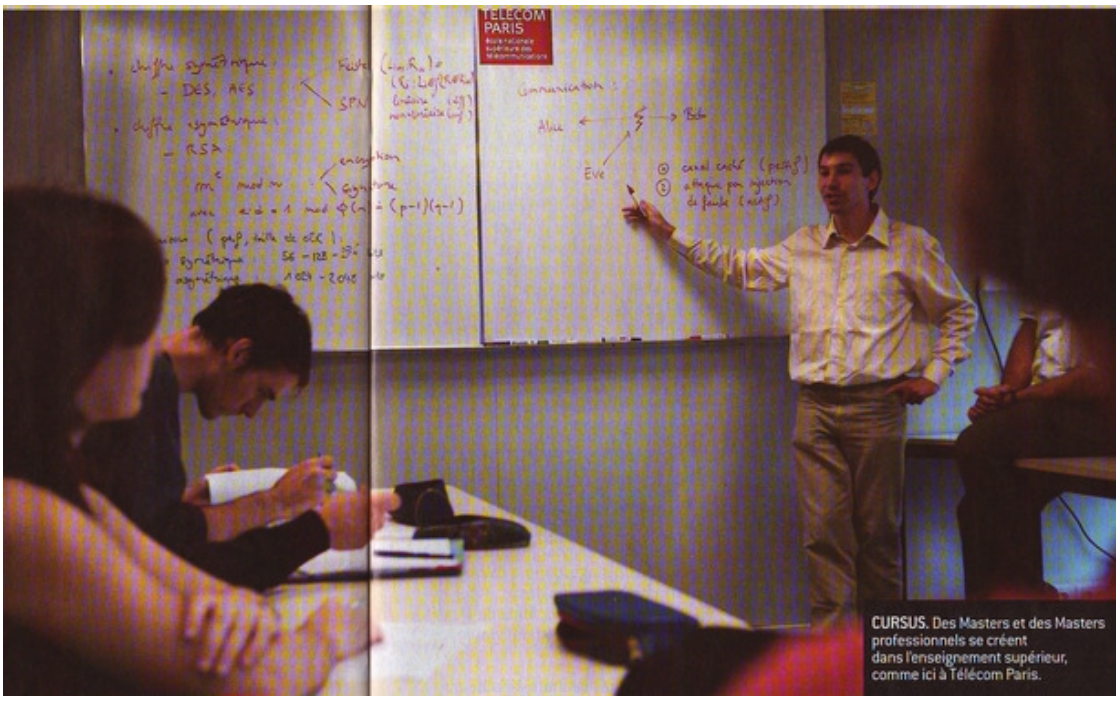


FIGURE 2.7 – Magazine `01Informatique`, août 2006.

FIGURE 2.8 – Magazine `01Informatique`, juin 2006.

Point de vue sur...
la gestion des droits numériques

Le terme de "société de l'information" est aujourd'hui entré dans le vocabulaire courant. On sait que l'informatique permet d'améliorer et d'automatiser beaucoup de services traditionnellement laborieux. Néanmoins, il est délicat de dire dans quelle mesure nous appartenons à la société de l'information. Celle-ci connaîtra réellement son essor quand les biens numériques seront commercialisés sur un marché "numérique" à définir. Deux types d'acteurs se penchent sur la question de la création d'un marché numérique.

D'un côté, les industriels proposent des solutions techniques pour les droits sur les biens numériques (DRM, Digital Rights Management). De l'autre côté, la Commission européenne [1] travaille sur les aspects juridiques visant à assurer une juste répartition de la valeur ajoutée.

Cet article apporte un point de vue critique sur la stratégie marketing des industriels qui vendront les technologies DRM au grand public et sur la perception des enjeux des DRM par la Commission européenne (lire cette partie sur www.enst.fr/interne/telecomedia).

Vers des restrictions d'usage...

La société de l'information est un univers virtuel où les fichiers sont copiables et distribuables à un coût pratiquement nul. Cette vertu, qui constitue l'extraordinaire potentiel de l'ère numérique, est néanmoins un obstacle à la commercialisation des biens numériques, comme les contenus multimédias ou les logiciels. En effet, tout bien numérique est vendu assorti de droits limitant son utilisation conformément au contrat de vente. Tout comme il est interdit de copier un film loué sur DVD, il est raisonnable de penser qu'un bien numérique vendu depuis Internet soit lui aussi interdit de copie.

Le but des DRM est d'implémenter ces restrictions d'usage. Ils nécessitent des moyens techniques garantissant que les biens numériques sont utilisés conformément à leurs droits. Les industriels proposent une plate-forme matérielle sécurisée ("trusted computing" [2] ou TCPA) qui est garante de la bonne utilisation des biens numériques. Celle-ci est capable d'évaluer son environnement et de sauvegarder ses mesures de manière sécurisée. Couplé à un système d'exploitation lui aussi sécurisé (comme Windows associé à Palladium [3]), TCPA peut empêcher l'exécution de logiciels pressentis comme ne respectant pas les DRM. Il est alors impossible de lancer un logiciel pirate ou de sauvegarder un film acheté pour une seule visualisation.

Les DRM permettent donc d'instaurer un marché des biens numériques dépourvu de fraude et de proposer de nouvelles habitudes de consommation répondant à de nouveaux besoins.

... et une perte de pouvoir de chacun sur ses données

Cependant, les DRM impliquent une perte de pouvoir des utilisateurs sur leur ordinateur personnel (PC) et sur "leurs" données. Anticipant un rejet de principe du grand public, le consortium TCPA avance donc un autre argument de promotion; c'est le gain de sécurité apporté par TCPA qui est mis en avant.

Mais l'argument est fallacieux : si TCPA sert en effet à sécuriser le marché des biens numériques, il n'accroît nullement la sécurité de l'utilisateur particulier. Les problèmes de sécurité que celui-ci rencontre, virus, bug ou spam, ne peuvent pas être résolus par TCPA. Affirmer que TCPA sécurise la société de l'information est donc un artifice marketing pour faire entrer, peut-être à son insu mais avec son assentiment (car qui oserait refuser la sécurité ?), le consommateur dans le marché numérique.
D'un point de vue éthique, cette stratégie marketing a deux effets pervers. Elle contribue à désinformer davantage le grand public sur les réels enjeux de la sécurité, ce qui conduit à la fragilisation de la sécurité de la société de l'information. Mais surtout elle attribue à tort à la société de l'information une connotation négative : celle-ci est décrite comme un lieu de non-droit où le seul moyen d'assurer sa sécurité est de s'abriter dans la forteresse de son PC.

Sylvain Guilley

[1] Directive 97/0359 sur le droit d'auteur et les droits voisins dans la société de l'information - [2] http://www.trustedcomputing.org - [3] Communiqué de Microsoft sur Palladium (renommé "Next-Generation Secure Computing Base") : http://www.microsoft.com/presspass/features/2002/jul02/0724palladiumwp.asp

***DRM** : droits d'utilisation d'un bien numérique. Il s'agit d'une méta-information qui accompagne ce bien. La confiance dans le futur marché numérique nécessite le respect des DRM. Ceux-ci doivent donc être sécurisés afin de prévenir le piratage. Chez le consommateur, quiconque possédant un PC relié à Internet, la sécurité des DRM dépend de celle des autres couches : matérielle (processeur) et système d'exploitation (OS). **TCPA** [2] : standard d'architecture matérielle sécurisée s'appuyant sur des techniques cryptographiques. Palladium ou NGSCB [3] est un module de l'OS Windows de Microsoft en charge d'assurer la continuité de la sécurité du matériel jusqu'aux applications.*

FIGURE 2.9 – Revue TELECOMEDIA, encart sur les droits de gestion numérique.

- **TOISE**, *Trusted Computing for European Embedded Systems*, projet ENIAC-2010-1, avec Thales, Gemalto, EADS, Cassidian, TST, CNM, ST, EAB, Magillem, DEA, Azcom, numonyx, BICOCCA, Proton, ICCS, Politechnico de Milano, CEA, Secure-IC.
  ⇝ http://www.toise.eu/
- **MARSHAL+**, *Mechanisms Against Reverse-engineering for Secure Hardware and Algorithms*, projet FUI 12, co-labellisé par les pôles de compétitivité System@tic et SCS. Les partenaires sont EADS, Secure-IC, INVIA, Inside-Secure, TRANEF, CryptoExperts, IRPI, Labri, UNILIM, UVSQ.
  ⇝ http://trac.marshalproject.org/
- **BMOS**, *Biometric Match-on-card System*, ANR « Systèmes Embarqués et Grandes Infrastructures » édition 2010, avec Morpho et Secure-IC.
  ⇝ https://bmos.enst.fr/
- **SPACES**, *Security evaluation of Physically Attacked Cryptoprocessors in Embedded Systems*, projet franco-japonais (ANR–JST), avec le LIP6, Morpho, Secure-IC, UEC, Tohoku U., AIST.
  ⇝ https://spaces.enst.fr/spaces
- **BCDL**, *Balanced Cell-based Dual-rail Logic*, projet RAPID DGA/DGCIS 2010, avec Secure-IC.
- **SecReSoc**, *Secured Reconfigurable System-on-Chip*, projet ANR du programme ARPEGE, projet ANR-09-SEGI-013, avec UHC, UBS, LIRMM, Netheos.
  ⇝ http://labh-curien.univ-st-etienne.fr/secresoc/doku_wiki/doku.php
- **SeFPGA**, *Secure embedded Field Programmable Gates Array*, projet ARFU labélisé par le pôle de compétitivité System@tic
  ⇝ https://sefpga.enst.fr/
- **Secure-Algorithms**, projet FUI 5 du pôle System@tic, avec Oberthur Technologies, Nagra, Thales, l'Université Paris 8, l'UVSQ.
  ⇝ https://secalgo.enst.fr/
- **HQ-NET**, *High bit-rate and versatile Quantum-secured NETwork*, projet ANR, avec SmartQuantum, Photline, UFC FEMPTO, UMI GeorgiaTech-CNRS.
  ⇝ http://hqnet.enst.fr/
- **EPOMI**, *Evaluation Plateforme Ouverte Modulaire & Incrémentale*, projet FUI, avec SFR, Orange, Gemalto, Oberthur CS, SAGEM/ORGA, TRUSTED LABS, CREDIT MUTUEL, RATP, GALITT, SERMA, DCSSI.
  ⇝ https://epomi.rd.francetelecom.com/public
- **CALISSON**, *CAractérisation, modéLIsation et Spécifications Sécuritaires de circuits prOtotypes iNtégrés*, projet FUI, avec ENSMSE, CEA/LETI/LCCS, TIMA, STM, PSI, Gemalto, Atmel.
  ⇝ https://tokyo.emse.fr/trac/calisson/
- **SAFE**, *Secured Asynchronous FPGA for Embedded systems*, ACI SI, avec TIMA.
  ⇝ http://projects.comelec.enst.fr/safe/
- **Conventions PACALAB**, sous-projet PS15, avec STMicroelectronics Rousset, département AST.

– **MARS**, *Matériel Robuste pour systèmes Sûrs*, ARA SSIA, avec TIMA.
  ⇝ http://projects.comelec.enst.fr/mars/
– **OpenSmartCard**, projet incitatif GET, avec l'ENST-Bretagne et la société JavaCard.
  ⇝ http://projects.comelec.enst.fr/opensmartcard/

2.11 Valorisation

J'ai participé à la création de la *spin-off* Secure-IC S.A.S., issue de l'essaimage de l'Institut TELECOM, qui valorise certains des brevets et des logiciels déposés par TELECOM-ParisTech. Cette société est implantée à Paris, à Rennes et à Singapour. L'implantation en Bretagne a permis de développer des liens de partenariats avec la DGA, notamment le centre de compétence « Maîtrise de l'Information », et les équipes sécurité – méthodes formelles de Télécom Bretagne (département Réseaux, sécurité et multimédia). Secure-IC a remporté le prestigieux concours national 2010 d'aide à la création d'entreprises de technologies innovantes du ministère de l'Enseignement supérieur et de la Recherche, catégorie « Création développement ». L'entreprise a également reçu d'autres distinctions, comme le prix Cré'acc 2010, le statut « EIP » du pôle de compétitivité System@tic ou l'électron d'or 2012 du magazine ElectronicS. En outre, la collaboration entre Secure-IC et TELECOM-ParisTech a été mise en avant par le CNRS (voir Fig. 2.10). Je sers Secure-IC en temps que conseiller aux conseils consultatifs scientifique et stratégique.

Design de circuits intégrés sécurisés

Description :

Secure-IC est spécialisée dans la protection des données des circuits électroniques, contribuant à la sécurité de systèmes informatiques afin de lutter contre les nouvelles techniques d'extraction des informations confidentielles. La société développe également une activité d'ingénierie et de conseil pour faciliter l'adoption de sa technologie ainsi qu'un outil d'analyse de la robustesse de circuits électroniques.

La technologie utilisée assure la confidentialité et l'intégrité des données matérielles et logicielles, notamment cryptographiques, incluant également la défense contre les attaques passives et actives, telles que les attaques par observation de canaux cachés et par injection de fautes. Elle améliore la résistance aux attaques des circuits électroniques, réduit leur consommation d'énergie ainsi que leur empreinte silicium.

Le produit phare de Secure-IC est un composant électronique cryptographique de type carte à puce de nouvelle génération ultra sécurisée, Smart SIC+, destiné à l'identification et qui devrait être commercialisé fin 2011.

Création : 28 Janvier 2010

Concours national d'aide à la création d'entreprises de technologies innovantes (2010)

Le produit d'analyse, Smart-SIC Analyzer, apporte, quant à lui, la quantification de fuite face à une attaque et ce sur n'importe quel type de design électronique (virtuel ou physique).

Les marchés visés par Secure-IC sont ceux de la défense et de la sécurité civile, pour les applications bancaires, documents d'identité électronique, cartes à puce, protection des communications, etc.

© SECURE-IC

Hassan TRIQUI, Président
contact@secure-ic.com

80, Avenue des buttes de Coësmes
35700 RENNES

http://www.secure-ic.com

Origine :

Secure-IC est née de la rencontre de M. Hassan TRIQUI, ancien directeur commercial chez Thales, Nextamp et Thomson avec le groupe « Systèmes Electroniques Numériques » du Laboratoire Traitement et Communication de l'Information de Paris (LTCI), unité mixte CNRS - Télécom ParisTech.
La société valorise des travaux de recherche conduits par MM. Jean-Luc DANGER, Laurent SAUVAGE et Sylvain GUILLEY, personnels de Télécom ParisTech au sein du département Communications et Electronique (COMELEC) du LTCI.

Laboratoire d'origine : UMR5141 – Laboratoire traitement et communication de l'information de Paris (LTCI)
Instituts : INS2I, INSIS
Délégation Régionale : DR01 – Paris A
Partenaires académiques : Institut Télécom et CNRS
Quelques références :

- *Demande de brevet FR N°08 51904 du 25 mars 2008 intitulée «Procédé de protection de circuit de cryptographie programmable, et circuit protégé par un tel procédé» citant comme inventeurs : Jean-Luc DANGER, Sylvain GUILLEY et Philippe HOOGVORST*
- *Logiciel fpgasbox déposé à l'Agence de Protection des Programmes (APP) le 9 juillet 2008*

Relations avec ses partenaires académiques :

Secure-IC devrait exploiter des résultats issus des travaux du groupe « Systèmes Electroniques Numériques » dans le cadre d'une licence concédée par l'Institut Télécom et le CNRS. L'objet de cette licence inclura notamment les références susmentionnées.

La société bénéficiera du concours scientifique de MM. Jean-Luc DANGER, Laurent SAUVAGE et Sylvain GUILLEY, personnels de Télécom ParisTech.

FIGURE 2.10 – Communiqué du CNRS intitulé « *Design de circuits intégrés sécurisés* » (2010).

Chapitre 3

Annexe : articles joints

Cette annexe présente des développements plus étayés que ceux donnés dans le premier chapitre.

Nous commençons tout d'abord dans la section A par un tutoriel sur les attaques et les contremesures, publié récemment dans une conférence de conception électronique. Il introduit notamment un canevas permettant de comparer sur une base rigoureuse les contremesures de masquage avec celles basées sur une logique à double-rail.

Pour illustrer la richesse des types de contremesures, nous illustrons deux méthodes de protection originales. Elles sont des compromis, c'est-à-dire non idéales théoriquement, mais efficaces en pratique. De plus, elles laissent toutes deux la possibilité au concepteur de faire jouer le rapport coût / sécurité. Dans le marché des produits sécurisés, ce degré de liberté est essentiel, car la meilleure réponse à un besoin est celle qui délivre exactement (ni plus, ni moins) le niveau de sécurité attendu, et ce au prix le meilleur. La première contremesure consiste à augmenter les parties combinatoires de l'implémentation au détriment des parties séquentielles. Elle est détaillée à la section F ; il s'agit d'une illustration classique d'exploration architecturale, appliquée à la sécurité. La seconde est un schéma de masquage pouvant être parfait, mais qui est sciemment dégradé pour diminuer sa taille d'implémentation [334]. Elle est détaillée dans la section G ; on constate que les propriétés attendues du masquage impliquent un choix sur les masques qui s'exprime comme un problème classique de codes. Ce qui est particulièrement novateur dans cette publication, c'est la confrontation formelle d'une métrique de fuite et d'une métrique de coût.

De même, l'imaginativité de l'attaquant est grande. Nous donnons deux exemples d'attaques originales. La première consiste en un attaquant qui arrive à combiner plusieurs attaques pour arriver plus vite à ses fins. Cette étude, conduite comme le développement de deux études de cas, figure à la section H. Ce travail est d'ailleurs assez porteur, car nous avons récemment eu l'occasion de l'approfondir encore plus [423]. Dans un second temps, nous montrons que l'objectif de l'attaquant peut être notoirement différent des craintes du concepteur d'un système que ce dernier estime sécurisé. Par exemple, en implémentant des algorithmes secrets, le concepteur peut ressentir une impression de sécurité. Or, comme nous le montrons dans la section I, théorie et pratique à l'appui, les

techniques d'attaque sur les canaux auxiliaires peuvent être adaptées pour retrouver des algorithmes secrets.

Nous faisons ensuite part d'une démarche complète de conception et d'évaluation d'une contremesure. La logique de calcul est décrite dans la section B, puis son câblage à la section C. L'évaluation, comparative à une version sans contremesure et avec une contremesure plus faible (WDDL), est donnée dans la section D. L'impact précis de tous les raffinements possibles des contremesures « backend » est analysé dans la section E.

Les contremesures visant à rendre l'activité constante protègent non seulement contre les attaques en observation, mais aussi contre les attaques en perturbation. Cette découverte heureuse est détaillée à la section J, dans le cas d'une logique à double rail séparable (WDDL). De plus, nous avons remarqué qu'une amélioration de la contremesure contre les attaques en observation, consistant à éliminer la propagation anticipée, permet également de mieux résister aux attaques en injections de fautes. Ce raffinement, lui-même heureux, est analysé à la section K.

Enfin, nous montrons comment ces idées débouchent sur la sécurisation des implémentations par la résilience. Le concept de résilience face aux injections de fautes est décrit à la section L. Ensuite, une unification de la résilience face aux attaques actives et passives est proposée dans la section M. Ces travaux pionniers posent les bases à des contremesures moins coûteuses, à condition toutefois que les protocoles cryptographiques les supportent. Certainement des améliorations sont envisageables (*cf.* idées de [166, 179], sur lesquelles portent des études en cours).

Appendix A

Vade Mecum on Side-Channels Attacks and Countermeasures for the Designer and the Evaluator

Extended version of article [199]

Authors: Sylvain Guilley, Olivier Meynard, Maxime Nassar, Guillaume Duc, Philippe Hoogvorst, Houssem Maghrebi, Aziz Elaabid, Shivam Bhasin, Youssef Souissi, Nicolas Debande, Laurent Sauvage and Jean-Luc Danger

Abstract

Implementation-level attacks are nowadays well known and most designers of security embedded systems are aware of them. However, both the number of vulnerabilities and of protections have seriously grown since the first public reporting of these threats in 1996. It is thus difficult to assess the correct countermeasures association to cover all the possible attack paths. The goal of this paper is to give a clear picture of the possible adequation between actually risks and mitigation techniques. A specific focus is made on two protection techniques addressing primarily side-channel attacks: masking and hiding. For the first time, we provide with a way to estimate a tradeoff depending on the environmental conditions (amount of noise) and on the designer skills (ability to balance the design). This tradeoff is illustrated in a decision diagram, helpful for the security designer to justify choices and to account for the cost overhead.

Key words: Implementation-level attacks, side-channel attacks, hiding and masking, leakage metric, comparison of countermeasures, decision diagram for the designer.

A.1 Introduction

Systems that process sensitive information can be the target of malevolent attacks that aim at recovering secrets illegitimately. Cryptography is the science that attempts to make it impossible for an attacker to retrieve private information. Encryption algorithms are typically used to conceal secrets. As a mathematical discipline, cryptography however makes some assumptions: the attacker is only expected to interact with the system through its regular interfaces. Now, when the cryptography is implemented in an embedded system, it is seriously challenged by attacks that make practical attempts to access the secrets. This means that all classical sneak tricks to access forbidden goods are possible. They include for instance spying, torturing, reversing or altering. Those actions are commonly referred to as "physical attacks".

A wealth of such attacks has been described and conducted experimentally with success on systems that were otherwise believed secure from the sole cryptographic standpoint. The first physical attack to be published was the "timing attack", presented at the conference CRYPTO in 1996 [247]. In this attack, an adversary is able to recover a secret key employed in a signature algorithm by spying on the time it takes for the system to output its result. This exploit is a typical "side-channel attack", insofar as it is completely passive: the attacked system does not even realize it is being stolen its secret key. Other side-channel attacks have been reported since then, and their study has mobilized many researchers. Those attacks unfold in two stages: side-channel collection and side-channel analysis (often abridged SCA). Side-channel collection is a straightforward "metrology" step, whereas SCA requires sophisticated tools to be efficient. Both aspects are advancing rapidly, as attested for instance by the "DPA contest" competitions [447]. In fact, the versions 3 and 4 are taking place in parallel in 2011 and address respectively the progress in acquisition and analysis of side-channel emanations.

This article focuses more particularly on SCA, because concepts involved in SCA are rich, and side-channel attacks can be conducted on virtually any embedded systems. Indeed, side-channel attacks enjoy two favorable properties. First of all, side-channel measurement is non-invasive: it seldom requires to modify or probe into the design. Second, side-channel attacks are passive, and thus the system is not aware of his being attacked, thus cannot take reactive countermeasures. This makes those attacks extremely likely to be mounted by non-professionals, with a fair chance of success unless the system is strongly leakage-proof. Thus, symmetrically, interesting countermeasures have been devised. They should have the specificity of being proactive, as the design must suppose it is constantly under attack.

The rest of the paper is structured as follows. An overview of side-channel attacks and countermeasures is given in Sec. A.2. Then, a more detailed analysis of specific countermeasures is described. Sec. A.3, A.4 and A.5 address countermeasures against respectively timing attacks, simple and differential power analysis attacks. Conclusions are in Sec. A.6.

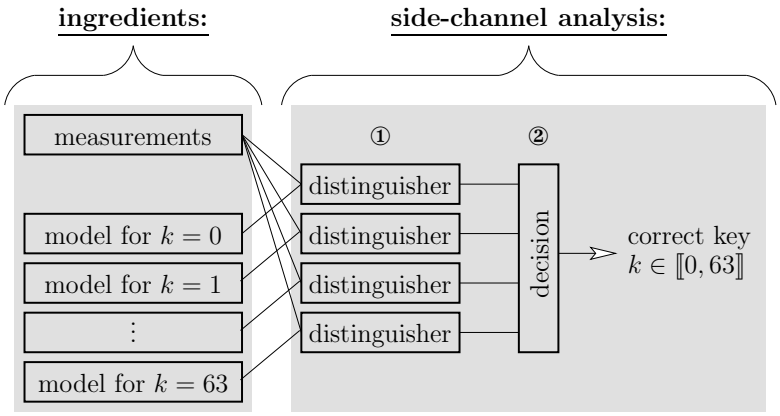


Figure A.1: Sketch of a side-channel attack where one correct key shall be extracted out of 64 key candidates.

A.2 Side-Channel Attacks and Countermeasures

A.2.1 Physical Side-Channels & Statistical Tools to Exploit Them

The side-channels can basically be sorted in two categories:

1. those where the duration of the cryptographic process is the leakage source, and
2. those where a physical quantity depending on time is leaked.

In the first case, for every invocation of the cryptographic primitive, a scalar is measured, whereas in the second case, many samples are collected. We call those samples a "trace", by reference to the name given to measurement files captured by digital oscilloscopes. The measured quantity can be for example the instant current drawn by the cryptographic device (power analysis [248, 290]) or the magnetic field it radiates (electromagnetic analysis [134]).

However, in both cases, the SCA unfolds according to a classical cryptanalytic scenario, that is depicted in Fig. A.1. The observations, either scalar or vectorial, are confronted to a model thanks to a distinguisher [89]. More precisely, as many models as secret key hypotheses are derived. In Fig. A.1, that applies to the case of DES key extraction, 64 models are considered. Indeed, in the DES algorithm (*confer* NIST FIPS PUB 46-3), each round key is consumed per words of 6 bits; guessing 6 bits of the first round key thus allows the attacker to predict 4 bits (because DES makes use of $6 \rightarrow 4$ substitution boxes) involved internally. The models can be any function of those four bits. Then, after the distinguisher has been applied, the attacker retains the most likely key.

Typically, the options for choosing a distinguisher are listed in Tab. A.1. For attacks that do not attempt to combine many samples from vectorial measurements, it has been argued in [291] that all these distinguishers are equivalent, *i.e.* that they eventually

provide the correct key and differ only by statistical deviations when the number of observations is insufficient.

A.2.2 Typical Attacks

Attacks can be divided into two categories, depending on the characteristic of the side-channel:

- **Simple** attacks consist in the direct analysis of the side-channel, which requires only one measurement per analysis.
- **Differential** attacks require many measurements to test one hypothesis on a secret.

Timing attacks are attacks where the side-channel is the computation duration. In practice, simple timing attacks do not exist. Indeed, a system that would have a response time that directly depends on the secret would be very badly designed (unless this behaviour is intentional). However, differential timing attacks have been described. They exploit the horizontal variations of a cryptographic process. For instance, in [247], Kocher *et al.* describe how an attacker can test the secret key bits of a remote server by comparing the time it takes to answer to a local simulation (same programme, same hardware).

The attacks that require traces use vectorial observations. The analyzed quantities are the traces vertical values. Under favorable experimental conditions, RSA can be analyzed using a single power trace. Indeed, if the two operations involved in the computations can be visually distinguished, the sequence of operations is revealed by only one trace. In this case, referred to as single power analysis (SPA), the attacks consist indeed in the analysis of simple vertical variations. In [239], Kasper *et al.* show how to break KeeLoq with SPA. Also, elliptic curve cryptography is especially vulnerable to both timing attacks and SPA, because the "double" and "add" operations in the inner iteration loop notably execute differently.

Differential vertical variations are exploited by the other attacks, when timing attacks and SPA are unpractical due to countermeasures. They consist in statistical extraction of the secrets based on the study of dependence between the observations and the models. The literature has studied many of them: all those listed in Tab. A.1 apply to SCAs taking advantage of differences in vertical variations (later on referred to as "DPA").

We provide in the code of Tab. A.2 an example of DPA using the Pearson linear correlation as a distinguisher. The example considers a key extraction from an acquisition campaign comprised of $10,000$ traces made up of $1,000$ samples each. The campaign is integrally saved in RAM in one matrix called `measurements`. The 2^6 models have been precomputed in variable `models`. The SCA itself consists in two steps, as already mentioned in Fig. A.1. The first step (①) is the evaluation of a distinguisher, whose result is stored in a $1,000 \times 64$ matrix, customarily called "differential traces". The second step (②) is the selection of the largest distinguisher value, which yields the correct key if the attack is successful.

It is not always trivial to define the most efficient attack. In this paper [348], authors mentioned that they succeeded in attacking KeeLoq in DPA when the algorithm was hardcoded. Now, when executed in software, the traces were misaligned due to a variable

Table A.1: Various distinguishers suitable for SCA.

Distinguisher	**Decision**	**Comments**
Difference of means (DoM)	Max.	Models are called "selection functions" [248]; refinements are provided in [309].
Covariance	Max.	Introduced initially as the multi-bit generalization of the DoM [28].
Correlation	Max.	Variants are Pearson [60] (often noted "ρ"), Spearman [21] or Kendall ("τ"), or Gini ("ξ") [423] correlation coefficients.
Likelihood	Max.	Used when probability density functions (PDF) can be estimated, and leads to Bayesian attacks [69].
Mutual information	Max.	Rely on off- or on-line PDF estimations [141, 281]. Models are also called "partitioning functions".
Kolmogorov-Smirnov	Max.	Rely on off- or on-line cumulative density functions (CDF) estimations [474, 285].
Least squares	Min.	Introduced in stochastic attacks [407, 406]. Winning distinguisher for the 1st DPA contest (by Ch. Clavier).
Variance	Min.	Many references are available [433, 274, 265, 219].
Principal components analysis (PCA)	Max.	First PCA (FPCA) [431] is a typical example of differential cluster analysis (DCA) [22].

Table A.2: Synoptic of a SCA in MATLAB. Other code examples can be found in the DPA Contest website [447] or in the `OpenSCA` [345] toolbox.

```
% Ingredients:
measurements = [[...];[...];[...]]; % Side-channel traces, 10000 x 1000 matrix
models       = [[...];[...];[...]]; % Models for all hypotheses, 10000 x 64 matrix

% Analysis:
distinguishers = corr( measurements, models, 'type', 'Pearson' ); % 1000 x 64 matrix
plot( distinguishers ); % Optional "sanity check" step, to see the 64 differential traces
[ maxcorr, maxindex ] = max( max( distinguishers )); % Decision function associated to corr
% The correct key is maxindex-1 (since in MATLAB, the indices start from 1 and not from 0),
% and corresponds to the greatest corr for all the 1000 dates & all the 64 key candidates.
```

duration of the encryption. Hence, an SPA happened to be the most efficient attack. In conclusion, the authors also note that timing attacks could be less error-prone than SPA on this device.

A.2.3 Provable Countermeasures: Information Masking or Hiding

In this article, we discuss so-called provable countermeasures. By provable, we assume two conditions: First all, the countermeasure must be sound, meaning that in the framework of a given model, it can be demonstrated that its principle do indeed protect efficiently. Second, it must adhere to Kerckhoffs' principle: it shall work even if its rational is completely exposed. Two counter-examples are for instance the dummy cycles insertion, since it is not sound [86], and the code obfuscation, since it involves a secret method that is not expected to hold long against a determined attacker.

The two provable examples we consider in the sequel are:

1. **information masking** [290, Chp. 9], which aims at randomizing the side-channel, and
2. **information hiding** [290, Chp. 7], which aims at balancing the side-channel.

A.3 Protection against Timing Attacks

A.3.1 Masking

Let us take the example of the computation of a modular exponentiation $M^d \bmod N$ of a message M to the power d modulo the RSA modulus N. To eliminate the derivation of links between d and the computation time of $M^d \bmod N$, one could think to take advantage of the following identity:

$$\left(M^{d_1} \bmod N\right) \cdot \left(M^{d_2} \bmod N\right) \equiv M^{d_1+d_2} \bmod N \ . \tag{A.1}$$

It makes a "secret splitting" strategy possible. At every RSA computation that involves private key d, the system draws a random number d_1, and derives d_2 such that $d = d_1 + d_2$. The computation time using Eqn. (A.1) now also depends on d_1, unknown to the attacker. Another masking countermeasure against timing attacks is called "secret blinding". For all random number r, we have: $M^{d+r\cdot\phi(N)} \equiv M^d \bmod N$. Hence a trivial way to randomize the execution length of RSA.

A.3.2 Hiding

The hiding countermeasure consists in having the computation unfold in a fixed amount of time. This solution works perfectly, because the timing is quantified (as clock periods). However, in practice, it is hard to really have a compiler produce portable and constant-time executables [247]. Hence assembly-level countermeasures, such as `xtime` for AES.

A.4 Protection against SPA

The protection of implementations against SPA requires greater skills than against timing attacks. Indeed, if the attacker has at her disposal a complete trace of execution, she can distinguish internal operations by their different timing if they leak information this way. We thus suppose as a pre-requisite that all key conditional operations execute in constant time.

A.4.1 Masking

The masking countermeasures presented against timing attacks do not apply to the protection against SPA. Indeed, let us assume internal operations can be distinguished via the observation of the side-channel [314]. Then the attacker retrieves d_1 and d_2 from implementations protected by exponent blinding, which trivially leads to $d = d_1 + d_2$. In the exponent splitting countermeasure, the attacker manages to extract $d + r \cdot \phi(N)$, that can be used as a legitimate private key.

Masking any internal operation seems very chancy. Thus, the protections against SPA rather rely on hiding.

A.4.2 Hiding

Basically, two approaches compete for the protection by hiding against SPA. The first one consists in having all the internal operations look similar. This is exemplified by the side-channel atomicity [81]. The second option is higher level. It aims at making the sequence of operations constant, using dummy operations (which proves to be dangerous, because of safe-errors [477]) or special redundant algorithms. For instance, the exponentiation based on the Montgomery ladder [234] also performs the same operations irrespective of the secret key.

A.5 Protection against DPA

A.5.1 Masking

Masking the operations consists in changing the representation of the sensitive data x, possibly each time they are used. This requires to find identities where the injected randomness m can be canceled out. Such identities are for instance:

1. $\forall m, (x \oplus m) \oplus m = x$, which gives rise to Boolean masking [148],
2. $\forall m \neq 0, (x \times m) \times m^{-1} = x$, which gives rise to arithmetic masking [6] (value 0 requires special care).

In these identities, x is the sensitive variable and m the random mask. If x is n-bit long, so is m. Other possibilities are affine masking [131], a combination of Boolean and arithmetic masking, and homographic masking [366].

Those countermeasures prevent first-order attacks, but still leak information. Therefore advanced attacks are possible. Notably, high-order attacks [307] exploit the residual leakage of masking schemes.

A.5.2 Hiding

The hiding countermeasure against DPA is predominantly implemented as dual-rail with precharge logic (*aka* DPL [96]). In this representation, every Boolean variable x is implemented as a couple of wires (x_t, x_f), such that:

- $(x_t, x_f) = (0, 0)$ or $(1, 1)$ in precharge phase, which prevents memory effects and enables positive (glitch-free [204]) computation, and
- $(x_t, x_f) = (x, \overline{x})$ in evaluation phase, which makes the activity independent of x.

This protection is easier to implement in hardware than in software. Indeed, in software, it is difficult to control the register transfers, all the more so as most of times, the internal architecture of the CPU is unknown. However, some works tend to show that hiding can be achieved in software too [220].

A.5.3 Comparison of Masking and Hiding against DPA

It is relatively easy and straightforward to get rid off design flaws that open the door to timing attacks and SPA. Now, fighting DPA is more difficult, and moreover, masking and hiding against DPA are costly countermeasures. It is thus important to compare them, because the designer has a major choice to make between them.

At first glance, masking seems easier to code properly, because it is a "source-level" countermeasure. However, if implemented at source-level, the masking is certainly doomed to fail. Indeed, a clever compiler will remove all the redundant data, and eventually end up with the optimized (and thus unprotected) description of the algorithm. Thus both masking and hiding schemes require writing the description of the countermeasure manually, at assembly language level for software or at netlist level for hardware.

In terms of area overhead, both masking and hiding require to duplicate the datapath. Variable x is represented as a masked variable and a mask in masking, and as a true

Table A.3: Illustration of the unbalance α on the resources' relative importance in the leakage.

Countermeasure	Resource	Weight	Leakage ($\mathcal{L}$)
Masking	n-bit mask	$1+\alpha$	$(1+\alpha)\cdot \mathrm{HW}(m)$
	n-bit masked data	1	$1\cdot \mathrm{HW}(x\oplus m)$
Hiding	n-bit true data	$1+\alpha$	$(1+\alpha)\cdot \mathrm{HW}(x)$
	n-bit false data	1	$1\cdot \mathrm{HW}(\overline{x})$

and a false variable in hiding. In terms of throughput, no change occurs for masking in hardware, since the masked data and the mask can be computed in parallel. By default, the throughput of DPL is halved with respect to the unprotected implementation, because of the precharge / evaluation sequence. However, some logic styles [331] manage to optimize this throughput by squeezing the precharge step. All in one, masking and hiding have a roughly comparable impact on the overhead.

Thus, to compare them, we consider only their level of security. The known flaw of masking is its susceptibility against high-order or information theoretic attacks, whereas hiding is rather susceptible to inaccurate balancing at the layout-level. To grasp both aspects, we introduce two parameters:

1. the amount of noise (assumed to be normally distributed) in the measurements, quantified by its variance σ^2, and
2. the backend unbalance, measured by α, a dimensionless parameter defined in Tab. A.3.

Ideal conditions for the defender correspond to $\sigma^2 = +\infty$ and $\alpha = 0$.

Hence the leakage models for n-bit resource x:

1. $\mathcal{L}_{\text{masking}}(x,m) \sim (1+\alpha)\cdot \mathrm{HW}(m) + \mathrm{HW}(x\oplus m) + \mathcal{N}(0,\sigma^2)$, where m is independent from x and follows a uniform distribution in $\{0,1\}^n$, and
2. $\mathcal{L}_{\text{hiding}}(x) \sim (1+\alpha)\cdot \mathrm{HW}(x) + \mathrm{HW}(\overline{x}) + \mathcal{N}(0,\sigma^2) = \alpha\cdot \mathrm{HW}(x) + \mathcal{N}(n,\sigma^2)$. This leakage model is optimistic: indeed, in practice, the bits are not likely to leak with the same unbalance. Rather, in a multi-bit context, we expect the unbalances to partially compensate one each other.

There are two kinds of security analyses that can be performed [434]. They lead to those metrics:

1. the success rate or the guessing entropy after an attack, and
2. the estimation of the leakage by information theoretic tools, such as the mutual information as a metric (MIM).

The first option is difficult, since masking and hiding countermeasures are not jeopardized by the same attacks. For instance, against first-order CPA [60], we have:

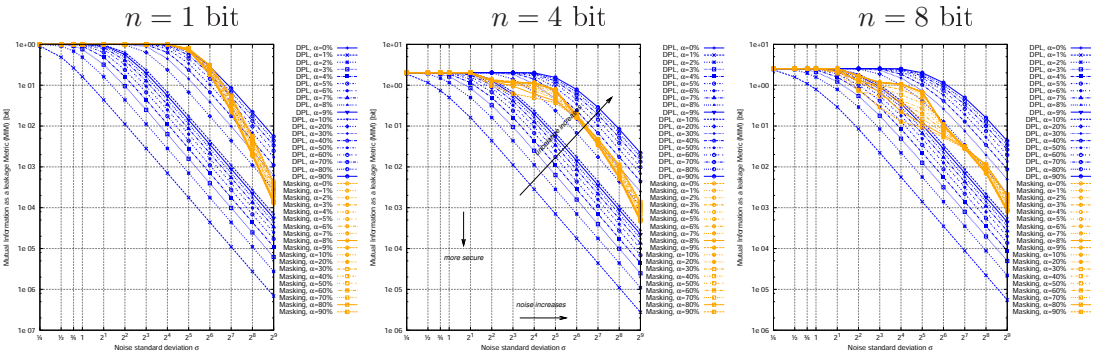


Figure A.2: Comparison between the leakage of DPL and masking countermeasures as a function of the experimental noise, for various α and for various n.

- $\rho_{x,m}\left(\mathcal{L}_{\text{masking}}(x,m); \text{HW}(x)\right) = 0, \forall \alpha$, whereas
- $\rho_x\left(\mathcal{L}_{\text{hiding}}(x); \text{HW}(x)\right) = \frac{\alpha\sqrt{n}}{\sqrt{n\alpha^2+4\sigma^2}} \neq 0 \quad \text{if} \quad \alpha \neq 0.$

Thus the information theoretic analysis is more suited in our case to compare the two countermeasures. Results are shown in Fig. A.2. It appears logically that the noise (quantified by its variance σ^2) reduces the mutual information, whereas the unbalance (quantified by α) increases it. However, the masking is much less impacted by the technological unbalance. The curves show that the less leaking countermeasure depends on the value of the couple (σ, α).

The leakage of the best countermeasure is plotted as a function of σ and α in Fig. A.3. The leakage is expressed in bits, and represented in logarithmic scale. The areas without color correspond to the equality between the two countermeasures. We see that, depending on the number of bits n considered in the analysis, the outcome changes. For the sake of illustration, we focus our analysis to the $n = 4$ case. It appears that, roughly speaking, for unbalances up to 17 %, DPL is the most secure choice. And for some values of the noise, namely $\sigma \in [2^4, 2^8]$, DPL remains the most secure solution for α up to 30 %. This graph therefore enables the designer to choose the most adequate countermeasure depending on the estimated environmental noise and on his ability to properly balance the layout.

A.5.4 General Picture

Before concluding, we wish to replace the problematic of protecting embedded systems into its general context. Side-channel attacks are only one class of attacks: what is thus the suitability of masking and hiding against the other attack strategies? The suitability of countermeasures to thwart attacks (as discussed in the previous paragraphs) is given in Fig. A.4.

This figure shows that masking is also a countermeasure against probing attacks, since the value of the probed node becomes random. Also, hiding is a countermeasure

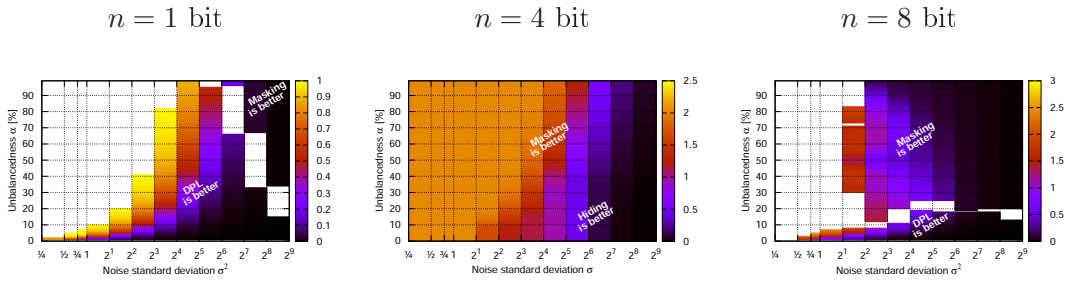


Figure A.3: Plot of domains where either masking or DPL leaks less (units: bit).

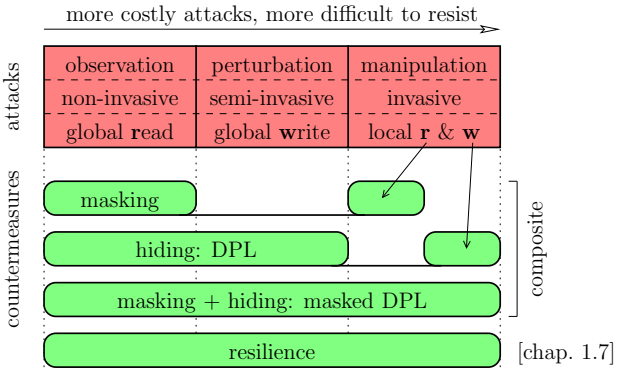


Figure A.4: Coverage of countermeasures for all physical attacks classes.

against most fault injection attacks since the attacker erases the value stored redundantly in one pair of wires by changing only one of them. The case of symmetric faults is covered in [206] and of arbitrary faults in [34].

An interesting noting is that by associating masking and hiding, the protection extends to semi-invasive and invasive attacks. This association must be realized with care, since otherwise some attacks become possible, such as the "folding attack" [405] or the "subset attack" [323]. The synopsis of this attack consists in recovering the masking bit and then to defeat the hiding countermeasure. However, by using more than one bit of mask, these attacks become impossible.

A.6 Conclusions

Cryptographic implementations can leak information in both time and amplitude. In this article, we provide a survey of known side-channels and we classify them according to their nature (horizontal / vertical) and the bias they disclose (simple / differential). Then, we review suitable countermeasures, and insist in particular on the masking and the hiding protection techniques. We specifically investigate these countermeasures in the context of vertical differential attacks, generically nicknamed DPA. It appears that they have roughly speaking the same cost, and thus differ only in the added security they bring to the design. We use a mutual information analysis to quantify their leakage, in the context of noisy measurements and imperfect resources matching. It appears that no countermeasure is better than the other in the complete studied domain. Instead, the choice depends on the environmental noise and on the skill of the designer to balance the resources at the backend-level. Eventually, we mention that masking and hiding can be constructively combined to achieve an immunity against all implementation-level attacks.

Appendix B

Security Evaluation of a Balanced Quasi-Delay Insensitive Library

Extended version of article [186]

Authors: *Sylvain Guilley, Florent Flament, Yves Mathieu and Renaud Pacalet*

Abstract

This article presents a library of cells enabling the realization of constant-power cryptoprocessors, natively protected against side-channel attacks. The proposed methodology uses a full-custom balanced quasi-delay insensitive (QDI) cell library, called "SecLib". It is suitable for a shielded routing method derived from the "backend duplication", using legacy CAD tools for the backend steps. The discussion is oriented towards the clarifying of topological constraints encountered in highly secure designs. We discuss the impact of intra-die technological mismatch on the security of SecLib.

Keywords: Standard cells design, power-constant logic, side-channel attacks mitigation, transistors mismatch, Monte-Carlo simulation.

B.1 Introduction

Side-channel attacks are a threat to the security of any electronic device. The seminal article of Paul Kocher [248] introduced several attacks, such as the SPA and especially the DPA, that can defeat cryptoprocessors, whatever the length of the keys. The vulnerability has been identified as an information leakage at the bit-level. Some high-level countermeasures against the DPA, such as duplicating [147] or masking [6], have been put forward. However, given the complexity of the underlying hardware, these solutions can be defeated by exploiting subtle non-logical phenomena, such as glitches [293].

Consequently, many *ad hoc* secured logic styles have been put forward. In the embedded security community, the so-called DPL (Dual-rail with Pre-charge Logic) family is overwhelmingly consensual. The DPL basically divide into two categories: "*power-constant*" and "*masked-power*" styles. In this paper, we investigate the feasibility of implementing optimally secured unmasked logic.

The rest of the paper is organized as follows. The specifications of the balanced QDI secured library "SecLib" is recalled in Sec. B.2. Then, the layout challenges of the secured logical gates design are dealt with in Sec. B.3. Finally, Sec. B.4 concludes the paper and provides some perspectives. The appendices B.5 and B.6 describe the derivation of SecLib gates respectively from a template in GDS2 to build the final gate layout and from a template in VHDL to build the final simulation model.

B.2 Specifications of SecLib

As the "SecLib" cell library is already extensively described by Guilley *et al.* in [188], only the prominent features are recalled in this section. SecLib is intended to be compatible, in terms of placement sites, with standard cells. This interoperability enables to reuse legacy cells for non-functional instances, such as scannable flip-flops, buffers, clock-gating logic, PN diodes and filler cells. SecLib, like other DPL libraries tailored for highly secured implementations, features security counter-measures at various levels: protocol, architecture, backend.

At the protocol level, a four-phase protocol enables to divide the computations into two steps: the computation proper and the precharge of the netlist. The first step consists in the computation of one iteration, while the second re-initializes all the nets so that the circuit is ready to start a new computation afresh, for instance with all the nets in a same electrical state.

Additionally, most secured cells rely on a dual-rail encoding: every logical bit is in fact carried by two wires. Many representations exist; however, a common one consists simply in associating the value *false* (0) to a wire and the value *true* (1) to the other. The rationale is to make any transition on the two wires indiscernible.

In dual-rail, every Boolean variable A is represented by a couple of two wires (A_0, A_1); when A is valid, $A = 0 \Leftrightarrow (A_0, A_1) = (1, 0)$ and $A = 1 \Leftrightarrow (A_0, A_1) = (0, 1)$. When A is invalid, $A_0 = A_1$. SecLib is optimized for $A_0 = A_1 = 0$.

The overall architecture of a representative SecLib gate (Fig. B.1) is classical to the QDI logic [193]. The inputs synchronization disables anticipated evaluation. The gate timing is thus unconditional to the data. This feature protects the gate against the signature differences of unsynchronized DPL caused by variations of input delay time [439]. the inputs configuration decoding $(A, B) \mapsto (C_{00}, C_{01}, C_{10}, C_{11})$ is well suited for an indiscernible processing. Notice that, for unbalanced functions, the computation part is forced to be symmetric by the use of dummy gates (*cf.* Fig. B.1 schematic on the *right*). SecLib is close to the logic described in this patent [116]; however, as shown in the sequel, SecLib is much easier to design and to dimension electrically due to the absence of bidirectional signals.

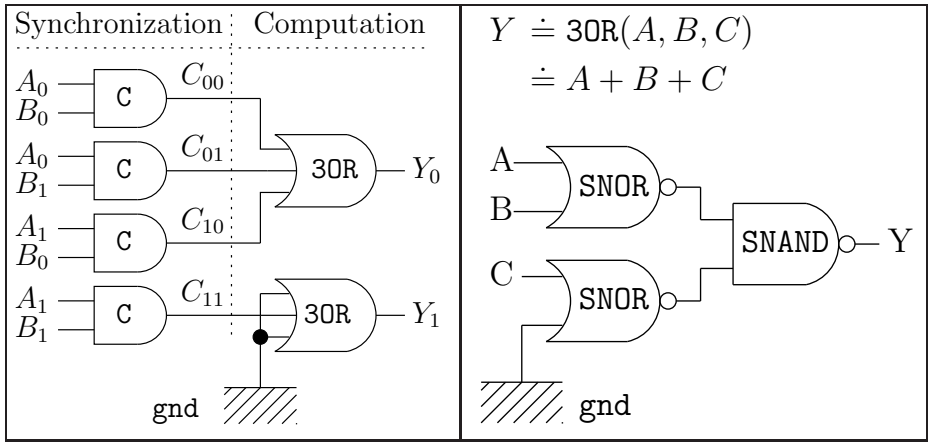


Figure B.1: Schematic of the QDI secured AND gate (*left*) and its internal 3OR architecture (*right*).

B.3 Layout of SecLib

B.3.1 Topological Issues Encountered in the Layout of SecLib

This section analyzes topological issues met when designing a library of dual-rail secured cells. It details the layout requirements arising from the *true* $\leftrightarrow$ *false* symmetry need. The layout issues can be circumvented to the sole SecLib instances, since non-functional gates (based on standard cells) do not leak any information. All layouts are realized in a 130 nanometers technology.

The structure of a balanced NOR (called SNOR, for Secured NOR) is shown in Fig. B.2(c). The layout challenge consists in porting the symmetry from the schematic to the masks. The basic steps are illustrated in Fig. B.2. First of all, an half-gate is designed (a). Then, two halves are instantiated, one in regular orientation R0, and the other in the mirrored orientation MY (b). This transformation allows for respect of an axial symmetry (the axis is denoted $\overrightarrow{\Delta}$.) The last step, (b) $\rightarrow$ (c), consists in the inner routing. It raises a topological problem, illustrated in Fig. B.3. It is impossible to connect the couples (A, A') and (B, B') without a short-circuit, which results in a functionally invalid solution. This concern is not specific to SecLib cells, but indeed inherent to any geometrical balancing strategy.

An approximation is provided with in Fig. B.4. Minimum sized polysilicium segments (130 nm $\times$ 180 nm), pointed out by arrows, connect the opposite nets: they are selected in Fig. B.4 (c). Those four segments constitute the sole symmetry violation.

The symmetrization methods presented above share the good property that transistors are paired in the same direction. This reduces the devices mismatches in case of mask misalignments during the manufacturing.

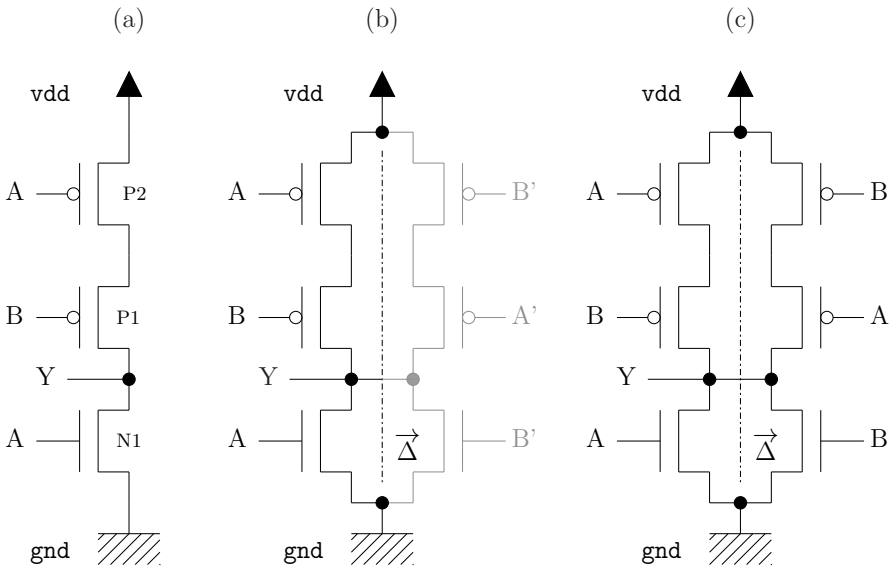


Figure B.2: Transistor-level schematic of a `SNOR` gate.

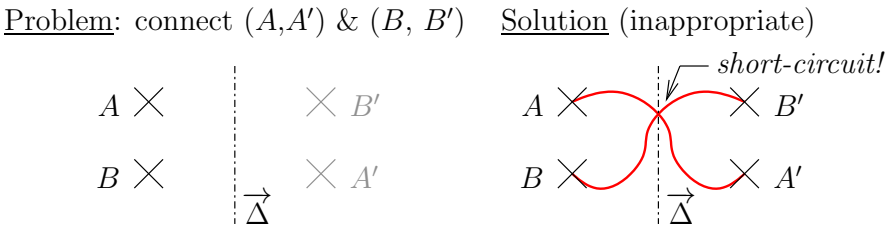


Figure B.3: $\overrightarrow{\Delta}$-symmetry topological problem (*left*); invalid solution (*right*).

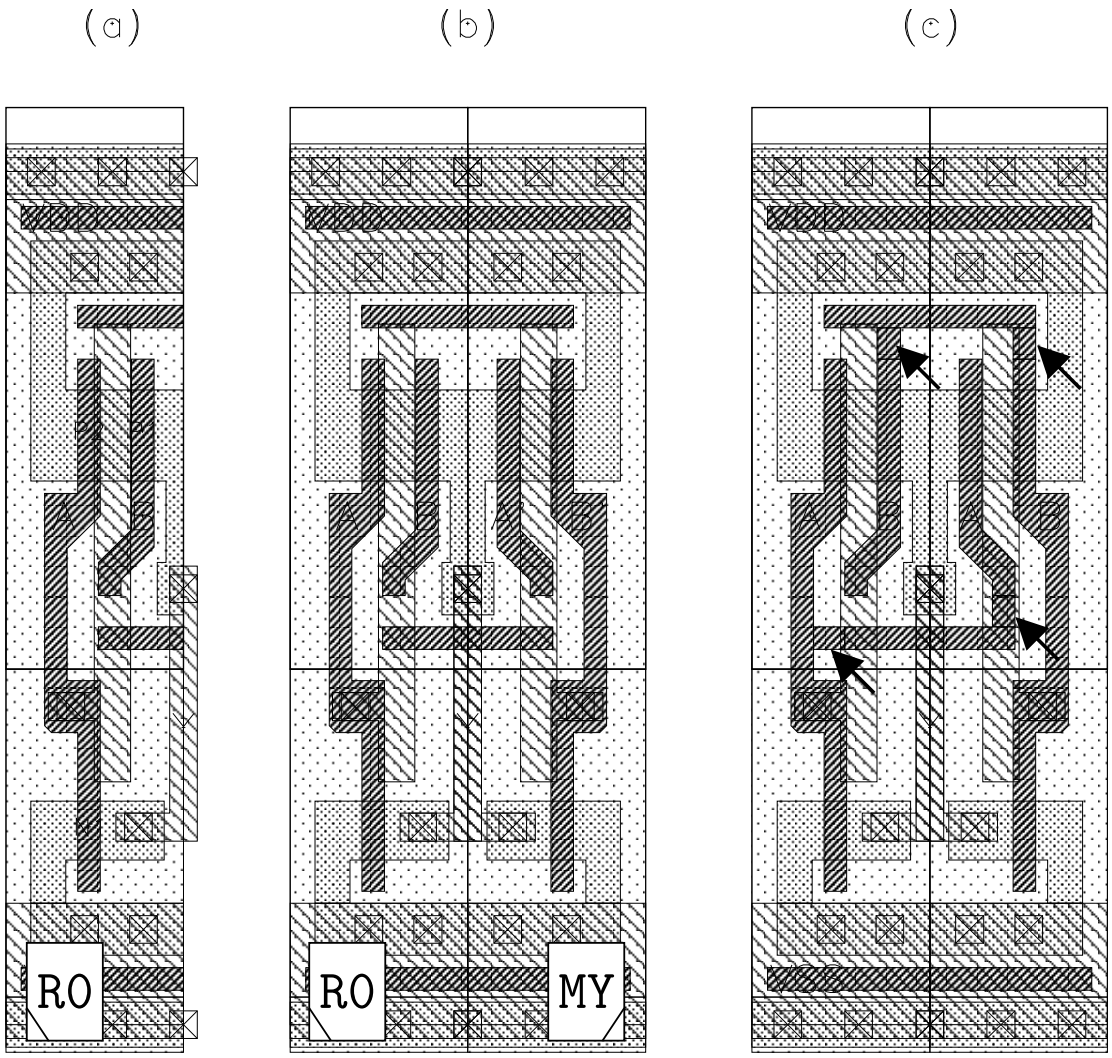


Figure B.4: Construction of a quasi-symmetric `SNOR` gate layout (*cf.* corresponding schematic in Fig. B.2).

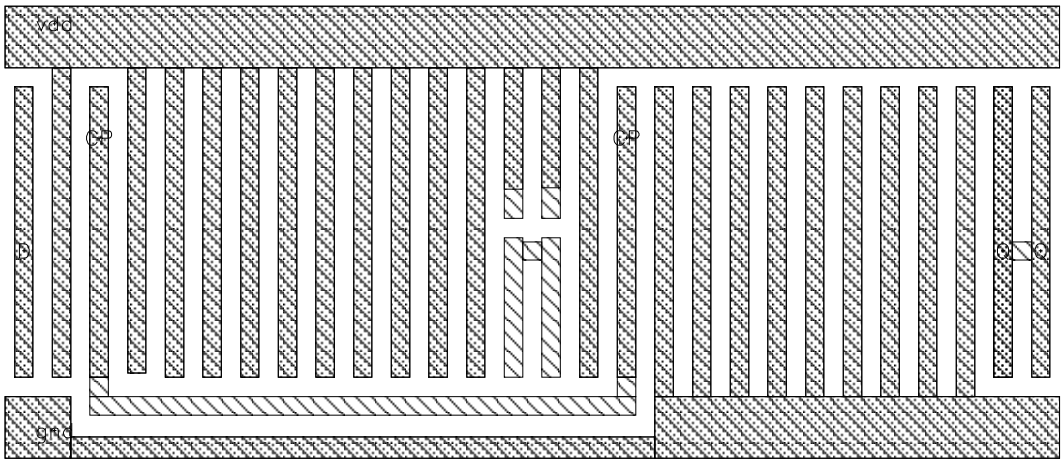


Figure B.5: Illustration of the M2 cage, on a D-flip-flop. D and Q pins are made available respectively on the left and right sides of the cell.

B.3.2 Gate Cocooning

A good cells library is geared towards the routability: the minimum number of metal layers must be used for the internal interconnections. In SecLib, only metals 1 and 2 are reserved for inner routing.

At the backend level, the decoupling between the computing logic and the routing resources is achieved thanks to an imprisonment of the transistors and the local interconnect in a `gnd`/`vdd` cage. The power/ground cage, illustrated in Fig. B.5, also provides two interesting benefits. First of all, the cell is a cocoon, where the computation takes place confidentially. The symmetry violation between the cell (*axial symmetry*, hence *odd*) and the routing (*translation*, hence *even* [188]) is thus minimized. Second, the cage is very convenient to connect the cell to the power and ground global nets. In Fig. B.5, the metal 2 pins (positive clock CP, input D, output Q, ground `gnd` and power `vdd`) are in bright cyan (▨), whereas obstructions for local interconnect are in low-intensity cyan (▧).

B.3.3 SecLib Gates Interfaces

The position and the shape of the pins is an important issue: in order to be visible from a differential pair, the pins must often be larger than expected. For instance, to comply with the "backend duplication" routing method [192], the pins must respect a vertical symmetry, which increases their extension.

This constraint arises from the conjunction of the two symmetries:

1. translation $T_{\overrightarrow{v}}$ by a vector $\overrightarrow{v}$ for the routing (upper constraint) and
2. glide reflection $S_{\overrightarrow{\Delta}}$ around an axis $\overrightarrow{\Delta}$ for the cell two halves (lower constraint),

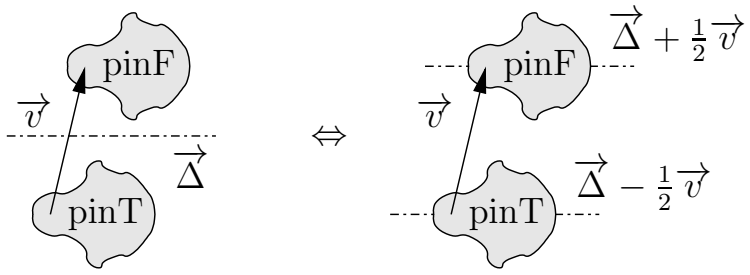


Figure B.6: Translation $\mathrm{T}_{\overrightarrow{v}}$ and reflection $\mathrm{S}_{\overrightarrow{\Delta}}$ symmetries to be met by dual pins.

that must be met concomitantly by the pins, because they constitute the interface between the two symmetries domains. More formally, if pinF (resp. pinT) is the set of points from the floorplan (*i.e.* in $\mathbb{R}^2$) that belong to the *false* (resp. *true*) pin, then the symmetries impose that:

$$\begin{cases} \text{pinF} = \mathrm{T}_{+\overrightarrow{v}}(\text{pinT}) & \text{(routing)} \\ \text{pinF} = \mathrm{T}_{+v_X\overrightarrow{e_X}} \circ \mathrm{S}_{\overrightarrow{\Delta}}(\text{pinT}) & \text{(cell)} \end{cases}$$

and reciprocally, that:

$$\begin{cases} \text{pinT} = \mathrm{T}_{-\overrightarrow{v}}(\text{pinF}) & \text{(routing)} \\ \text{pinT} = \mathrm{T}_{-v_X\overrightarrow{e_X}} \circ \mathrm{S}_{\overrightarrow{\Delta}}(\text{pinF}) & \text{(cell)} \end{cases}$$

The second constraint can be simplified as the following local constraints:

$$\begin{cases} \text{pinF} = \mathrm{S}_{\overrightarrow{\Delta}+\frac{1}{2}v_Y\overrightarrow{e_Y}}(\text{pinF}) & \text{(pinF symmetry)} \\ \text{pinT} = \mathrm{S}_{\overrightarrow{\Delta}-\frac{1}{2}v_Y\overrightarrow{e_Y}}(\text{pinT}) & \text{(pinT symmetry)} \end{cases}$$

The proof is given below for pinT (the demonstration for pinF is much similar):

$$\begin{array}{l} \forall (x,y) \in \text{pinT}, (x',y') = (x - v_X, y - v_Y) \in \text{pinF}, \\ \text{thus } (x'',y'') = (x' + v_X, 2 \cdot \Delta_Y - y') = \\ (x, 2 \cdot (\Delta_Y - \frac{1}{2}v_Y) - y) \in \text{pinT}\,. \end{array}$$

Figure B.6 illustrates this "symmetry transportation" result.

Whenever possible, the pins are placed on the cell right and/or left sides so that two neighbor cells can be routed directly in metal 2. These recommendations are applied on SecLib gates, as shown on the example of the SecLib `AND` instance in Fig. B.7.

The layout of other 2-input gates can be transposed straightforwardly from that of the `AND` gate. For instance, the family $(A,B) \mapsto \{\bar{A} \cdot \bar{B},\ \bar{A} \cdot B,\ A \cdot \bar{B},\ A \cdot B\}$ can be drawn based on the same *template*, specialized by the addition of vias at the relevant places [162]. Some details are provided in appendices B.5 and B.6. SecLib cells

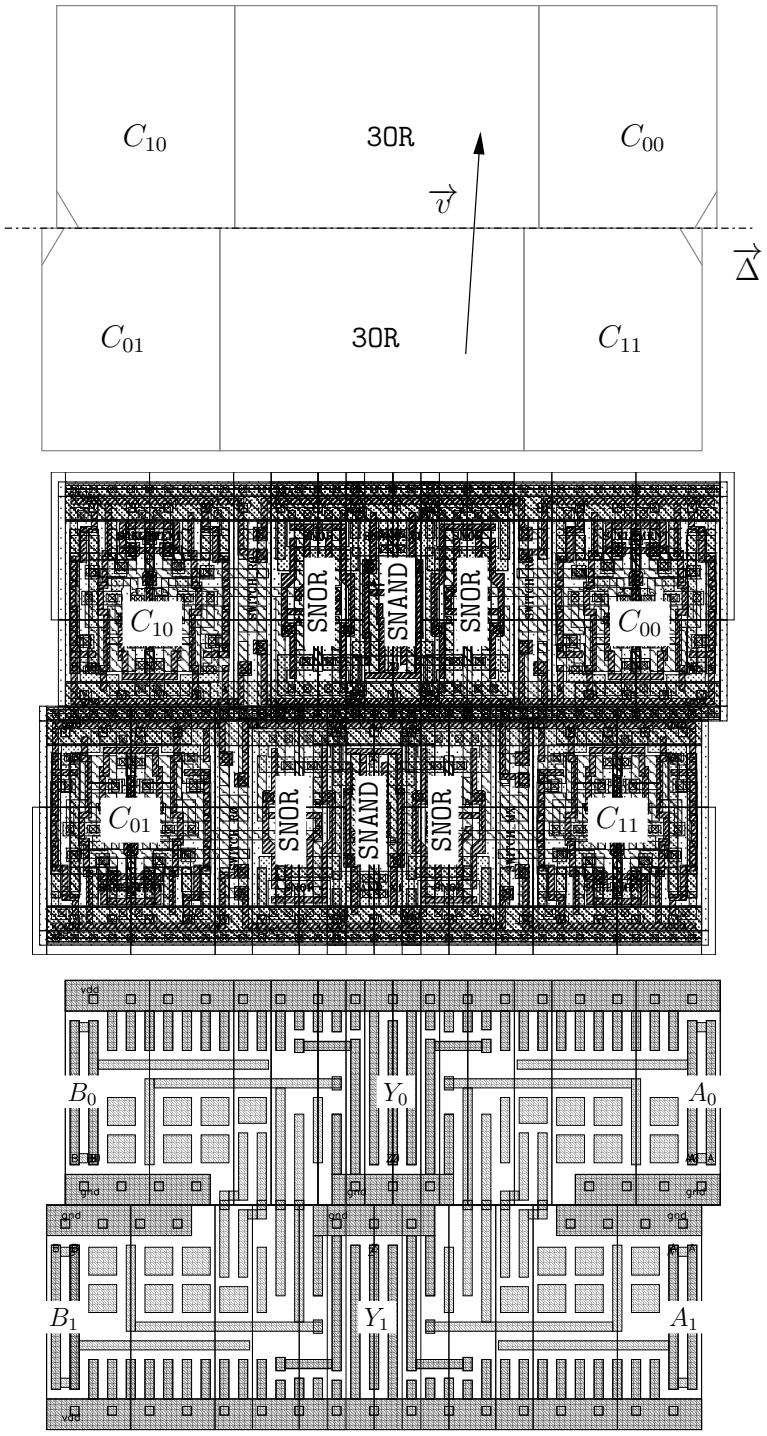


Figure B.7: SecLib two-input AND gate floorplan (*top*), structure (*middle*) and interface (*bottom*).

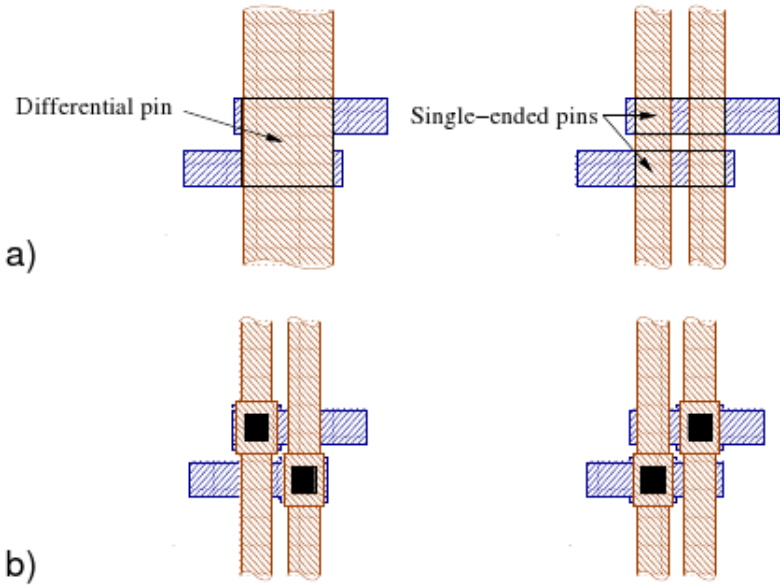


Figure B.8: Differential pin placement compatible with the fat-wire routing: a) Creation of individual pins out of a "virtual" differential pin. b) Two possibilities of placing the contacts. (*The figure 2 of [15] is reproduced here*)

are asynchronous, hence hazard-free: arbitrary Boolean functions can be implemented. Other non-synchronizing logics must restrict themselves to positive functions in order not to generate and not to propagate data-dependent glitches. The average density of SecLib is 545 527 transistors/mm^2, versus 766 586 for the standard cells.

Incidentally, we note that the question of pin design for dual-rail logic with fat-wire [457] routing is addressed in [15]. Basically, the dual pins are localed on two diagonally adjacent placement sites, so as to contact to wires on two neighbor routing tracks. The figure B.8 represents the possible connexion configurations.

B.3.4 Mismatch Impact on Gates Balancedness

In deep sub-micron technologies, the electrical parameters are subject to local mismatches, that potentially wreak havoc the symmetry of secured gates. The term *mismatch* is defined as the electrical parameter deviation between identically designed components. It is customarily used in analog devices to predict their unbalancedness. The mismatch results from electrical fluctuations induced by nanoscopic variations in physical quantities.

A study on the mismatch in a differential interconnect network is carried out in [227]. This sub-section accounts for the threshold voltage mismatch simulation on the instant and average current consumed by secured DPL gates. Both SecLib and WDDL [456] logics are studied, based on the example of an `AND` gate. The comparison is made be-

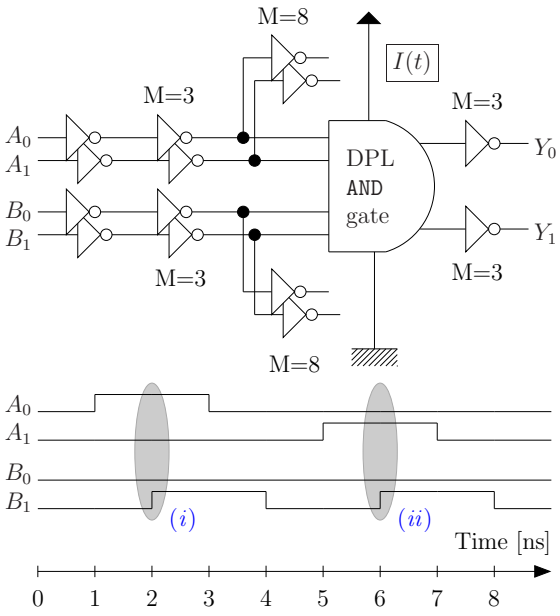


Figure B.9: SPICE testbench for DPL gates instant current $I(t)$ extraction.

tween those two logic styles because they both use "full-amplitude" signals (from `gnd` to `vdd` volts – as the standard cells provided in founders design kits), which would not be the case for SABL [452] for instance. The testbench is depicted in Fig. B.9. The environment is comprised of unitary inverters, of various multiplicities (M=3 or M=8): these values are chosen because they are representative of typical gates neighborhood. The DPL gate is powered by a separate supply, whose current $I(t)$ is extracted. Transistors are provided in 130 nm technology with mismatch models based on Pelgrom's linear characterization [354]. The Monte-Carlo option of electrical simulators is used to launch 500 simulations. The waveforms are represented in Fig. B.10 for SecLib and WDDL logics.

The relative difference of the instant current $I(t)$ and of the integrated current $\int I(t)\,\mathrm{d}t$ over the transition length are computed between: (i) the transition $A = 0$, $B : 0 \to 1$, and (ii) the transition $A = 1$, $B : 0 \to 1$. This relative difference between these two events is chosen because it is representative of the average unbalancedness that an attacker might exploit. The results are summarized in Tab. B.1 in the form: "mean $\pm$ standard deviation", expressed in percent.

The dispersion is important (about 5 %) on the maximum current peak amplitude. This figure is trustworthy, since commensurate with empirical estimations carried out on a similar technology (90 nm instead of 130 nm) [409]. The mean relative difference is masked in the standard deviation for both SecLib and WDDL. The standard deviation is greater for SecLib, because the gates belonging to this library are comprised of more transistors than WDDL ones. The statistics on the average current relative difference

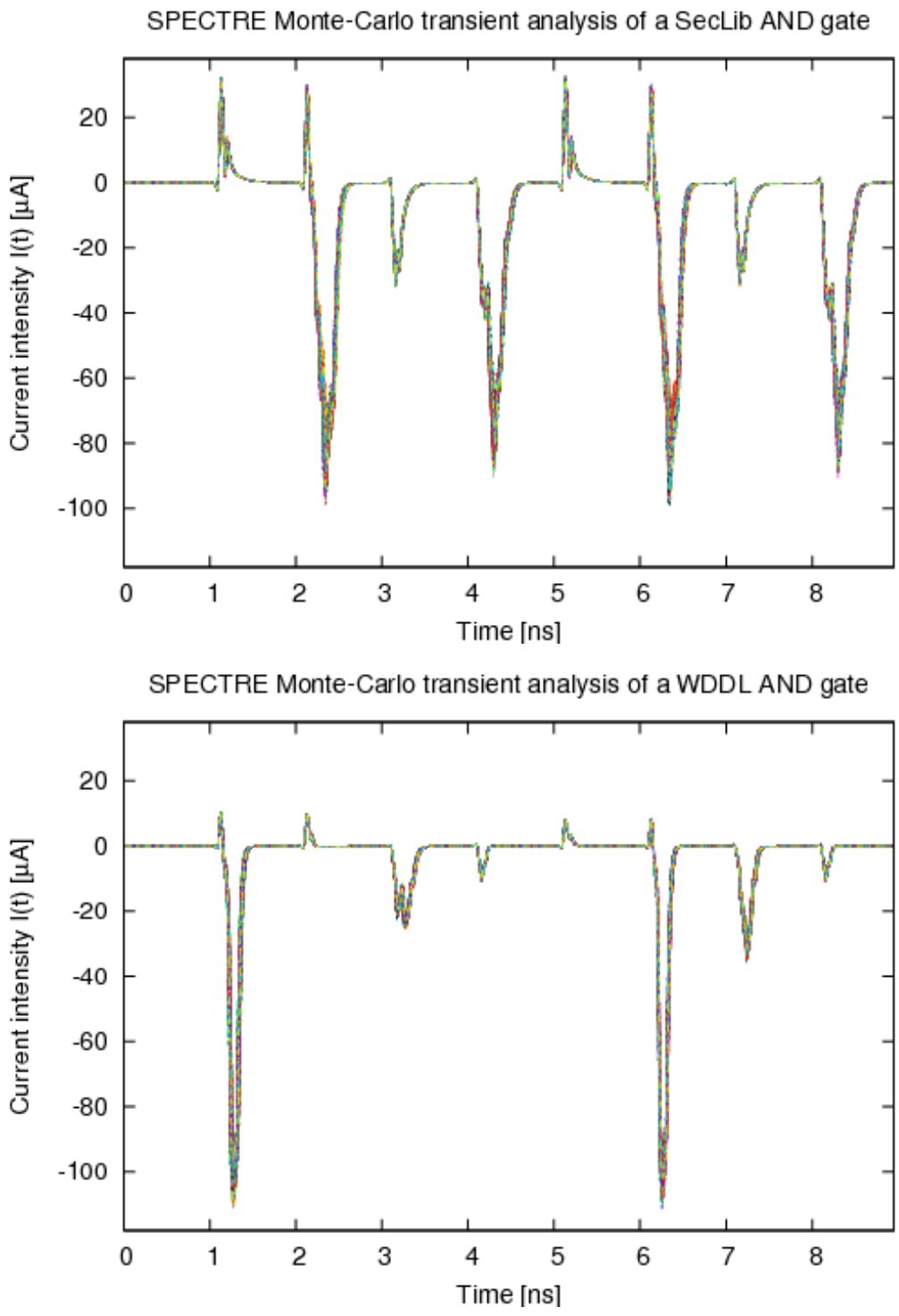


Figure B.10: Monte-Carlo simulation results for SecLib (*top*) and WDDL (*bottom*).

Table B.1: Relative difference of the maximum and the integrated current consumed by two DPL gates.

	SecLib	**WDDL**
$\max I(t)$	(-1.01 ± 5.46) %	(-0.36 ± 4.87) %
$\int I(t)\,\mathrm{d}t$	$(+0.01 \pm 0.33)$ %	$(+1.63 \pm 0.22)$ %

show that:

- SecLib is more balanced than WDDL ($|+0.01|$ % *versus* $|+1.63|$ %),
- the mismatch is the overwhelming source of unbalancedness for SecLib, because the standard deviation is much greater than the mean (0.33 % $\gg |+0.01|$ %),
- the structural unbalancedness of WDDL is the principal cause of its unbalancedness (0.22 % $\ll |+1.63|$ %).

The "integrated current" metric is believed to be the most representative of measurements that an attacker might realize concretely: as a matter of fact, every measurement is low-passed filtered, because of the on-chip power grid and of the on-package decoupling capacitances [261, p. 33]. In conclusion, simulations tend to show that, from the pure computational standpoint, the level of security of SecLib logic is limited by the mismatch, while WDDL is still limited by its intrinsic asymmetry.

B.4 Conclusion & Perspectives

This paper revisits the design of statically secured cells suitable for constant-power custom cryptographic ICs. Most previously proposed gates are vulnerable to a power attack exploiting the inputs skew. Therefore, this article focuses on a logic style (SecLib) in which gates inputs are systematically resynchronized. A method to port the symmetry constraints from the schematic to the layout is made explicit. We emphasize the topological issues raised by the symmetric routing constraints. The question of the positions of the pins is extensively discussed. This issue is indeed crucial since it allows the gates to support balanced differential routing. The paper concludes positively on the feasibility of industrial-strength secured cells libraries. One strong contribution of this paper is to show that secured logics based on standard cells, such as WDDL, are limited by the unbalanced design, but that the balancedness of SecLib is limited only by the intra-die technological mismatch.

Future works will focus on the study of sequential gates (such as memory elements) and of complex circuits (comprised of more than one single gate).

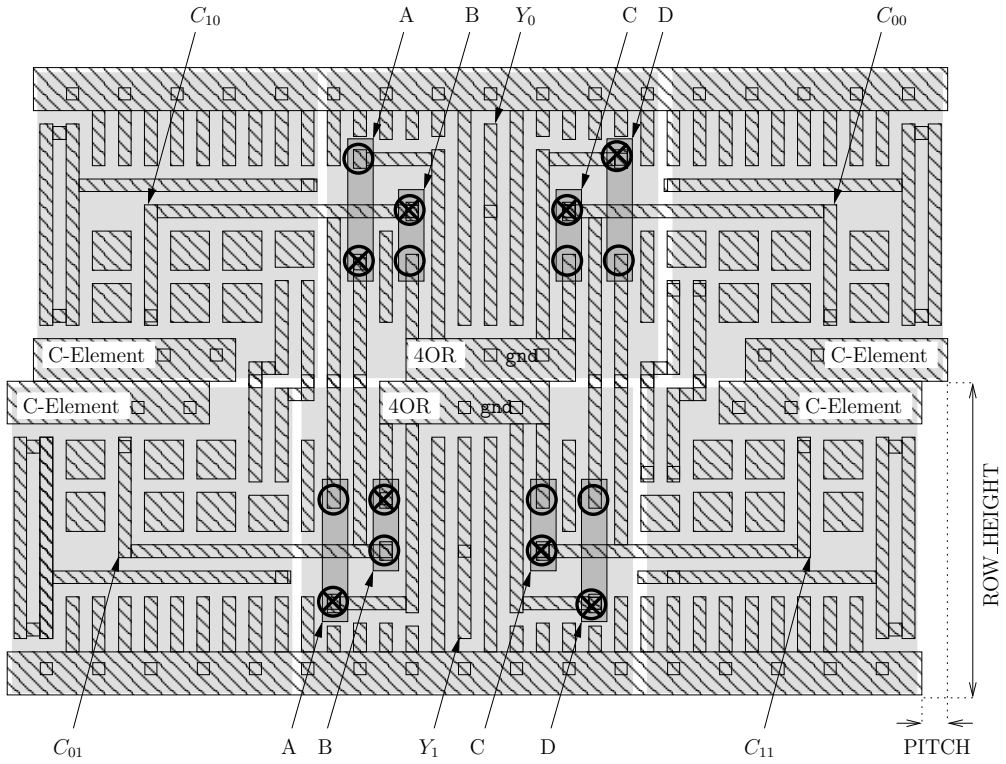


Figure B.11: Layout of an unfinished SecLib gate: vias can be added at positions spotted by circles.

B.5 Appendix 1: Generation of the Layout of Two-Input SecLib Gates from one Template

This section explains how to generate multiple two-input gates from one single GDS2 template. The structure given in Fig. B.1 involves a `3OR` gate. For symmetry reasons, this `3OR` gate is actually one `4OR` with one input shorted to the ground. We provide in Fig. B.11 with the layout (only metal-2 is shown) of an unbalanced (`XOR` and `XNOR` are excluded) two-input gate template.

The four inputs A, B, C & D of the `4OR` can be connected either to the ground or to one C-Element gate output. The connection points are indicated by a circle (◯) in Fig. B.11. One via is instantiated to choose the adequate connection; it is represented by a cross into the circle (⊗) in Fig. B.11. Table B.2 summarizes the connection possibilities; the vias are indicated by underlining the selected input for each input of the two `4OR` gates driving the differential output (Y_0, Y_1).

Table B.2: Connectivity within a two-input SecLib template gate to specialize it to a `AND`.

Input \ `4OR`	`4OR` **driving** Y_0	`4OR` **driving** Y_1
A	`gnd` *or* $\underline{C_{01}}$	$\underline{\texttt{gnd}}$ *or* C_{10}
B	`gnd` *or* $\underline{C_{10}}$	$\underline{\texttt{gnd}}$ *or* C_{01}
C	`gnd` *or* $\underline{C_{00}}$	`gnd` *or* $\underline{C_{11}}$
D	$\underline{\texttt{gnd}}$ *or* C_{11}	$\underline{\texttt{gnd}}$ *or* C_{00}

B.6 Appendix 2: Generation of the Behavioral Description of Two-Input SecLib Gates from one Template

The VHDL behavioral description of n-input QDI gates, upon which SecLib gates are built, is listed below. It enables fast functional simulations of SecLib netlists.

```
-- @file   qdi.vhd
-- @brief The behavioral specification of the quasi-delay insensitive (aka QDI)
--        primitives used in SecLib (Secured Library) logic.

library ieee;
use ieee.std_logic_1164.all;

-- Multiple input / single output 1-of-2 four-phase QDI gate behavioral model:
entity qdi is
  generic
  (
    tt: std_ulogic_vector -- Truth table
  );
  port
  (
    -- A(False, True), B(False, True), C(False, True) [if a'length = 6].
    a: in  std_ulogic_vector;
    y: out std_ulogic_vector
  );
begin
  assert a'length mod 2 = 0 and y'length = 2
  report "QDI gate ports are not dual-rail"
  severity failure;
  assert tt'length = 2**( a'length / 2 )
  report "QDI gate truth table has a bad dimension"
  severity failure;
end entity qdi;

architecture beh of qdi is
  signal c: std_ulogic_vector( 0 to 2**( a'length / 2 ) - 1 ); -- ''a'' decoded
  -- Evaluates whether one '1' of the truth table is hit. Models an OR gate:
  function eval( signal c: in std_ulogic_vector )
  return std_ulogic_vector is
    variable result: std_ulogic_vector( 0 to 1 ) := "00";
  begin
    for I in c'range loop
      if ( not tt( I ) and c( I )) = '1' then result( 0 ) := '1'; end if;
      if (     tt( I ) and c( I )) = '1' then result( 1 ) := '1'; end if;
    end loop;
    return result;
  end function eval;
begin
  -- Example on 3 bits:
  -- +---------------+---------+
  -- |               | A  B  C |
  -- |     0 1 2     | 01 01 01 | <= "01" means "True,False"
  -- +---------------+---------+
```

```
    -- | c( "0 0 0" ) | YN YN YN | <= Bits to test (Y=Yes, N=No) against 0 or 1
    -- | c( "0 0 1" ) | NY YN YN |
    -- | c( "0 1 0" ) | YN NY YN |
    -- | c( "0 1 1" ) | NY NY YN |
    -- | c( "1 0 0" ) | YN YN NY |
    -- | c( "1 0 1" ) | NY YN NY |
    -- | c( "1 1 0" ) | YN NY NY |
    -- | c( "1 1 1" ) | NY NY NY |
    -- +----------------+----------+
    G_DECODE: for C_I in c'range generate
      -- Testing concomitant ''all 0'' and ''all 1'' bits:
      P_SEQUENTIAL_C_I: process( a ) -- Models a C-Element with a'length/2 inputs
        -- The type ''boolean_vector'' does not exist in VHDL, unfortunately:
        variable rdv: bit_vector( 0 to 1 );
        -- Tests whether or not the bit at position ''pos'' of the (32-bit)
        -- integer is set.
        function is_set( a: integer; pos: natural ) return boolean is
        begin
          assert pos < 32 -- Portability notice
          report "Integers are often represented as 32-bit strings"
          severity warning;
          if ( a/2**pos mod 2 ) = 0
          then return false;
          else return true;
          end if;
        end function is_set;
        function is_set( a: integer; pos: natural ) return integer is
        begin
          if is_set( a, pos )
          then return 1;
          else return 0;
          end if;
        end function is_set;
      begin
        rdv := "11"; -- By default, a double RdV... now cancelling the bad choices
        for ABC in 0 to a'length / 2 -1 loop -- n iterations for n-input gates
          if A( 2 * ABC + is_set( C_I, ABC )) = '1' -- The bit is set
          then rdv( 0 ) := '0'; -- No rendez-vous to ''0''
          else rdv( 1 ) := '0'; -- No rendez-vous to ''1''
          end if;
        end loop; -- On A, B, C, etc. dual-rail signals concatenated in ''a''
        assert not( ( rdv( 0 ) and rdv( 1 ) )= '1' )
        report "One C-Element reported a rendez-vous to both ''0'' and ''1''"
        severity failure;
        -- Updating ''c'' only if there were actually a rendez-vous:
        if rdv( 0 ) = '1' then c( C_I ) <= '0'; end if;
        if rdv( 1 ) = '1' then c( C_I ) <= '1'; end if;
      end process P_SEQUENTIAL_C_I;
    end generate G_DECODE;

    P_OUTPUT: y <= eval( c );

  end architecture beh;
```

```
101
library ieee;
use ieee.std_logic_1164.all;
use work.all;

106 -- Two-input QDI gate
entity qdi2 is
  generic
  (
    tt: std_ulogic_vector -- Truth table
111 );
  port
  (
    -- a(False, True), b(False, True) => y(False, True)
    A0, A1, B0, B1: in  std_ulogic;
116 Z0, Z1:         out std_ulogic
  );
end entity qdi2;

architecture adaptor of qdi2 is
121 signal inputs:  std_ulogic_vector( 0 to 3 ); -- A_False A_True//B_False B_True
  signal outputs: std_ulogic_vector( 0 to 1 ); -- Z_False Z_True
begin
  P_INPUTS: inputs <= A0 & A1 & B0 & B1;
  I_QDI: entity qdi( beh )
126   generic map( tt => tt )
    port    map( a => inputs, y => outputs );
  P_OUTPUTS_Y0: Z0 <= outputs( 0 );
  P_OUTPUTS_Y1: Z1 <= outputs( 1 );
end architecture adaptor;
```

Finally, the behavioral description of the SecLib `AND` gate is given below:

```
library ieee;
use ieee.std_logic_1164.all;
use work.all; -- For the visibility of the previously described entity "qdi2"

 5 entity SAN2_X1 is
  port
  (
    A0, A1, B0, B1: in  std_ulogic;
    Z0, Z1:         out std_ulogic
10 );
end entity SAN2_X1;

architecture template of SAN2_X1 is
begin
15 I_QDI2: entity qdi2( adaptor )
    generic map( tt => "0001" )
    port    map( A0 => A0, A1 => A1, B0 => B0, B1 => B1, Z0 => Z0, Z1 => Z1 );
end architecture template;
```

Appendix C

Secured CAD Back-End Flow for Power-Analysis Resistant Cryptoprocessors

Extended version of article [187]

Authors: *Sylvain Guilley, Florent Flament, Philippe Hoogvorst, Renaud Pacalet and Yves Mathieu*

Abstract

This article presents a comprehensive backend design flow enabling the realization of constant-power cryptoprocessors, natively protected against side-channel attacks exploiting the instant power consumption. The proposed methodology is based on the use of a full-custom balanced cell library and on an innovative place-and-route method. The aim of this paper is to show that a piece of hardware, robust against all known power attacks, can indeed be implemented. All the design steps involved in the presented methodology take place at the layout level. The described flow has been applied to the quasi delay-insensitive "SecLib" library with a shielded routing method derived from the "backend duplication", using legacy CAD tools for the backend steps. The cost of the secured methodology is evaluated on the example of a multi-mode DES datapath; it appears that in deep sub-micron technologies, the design is in the *wire-domain*, *i.e.* limited by the interconnect resources.

Keywords:

Robust hardware, backend design automation, power-constant architectures, side-channel attacks mitigation, design for manufacturability and yield (DFM and DFY).

C.1 Introduction

Side-channel attacks are a threat to the security of any electronic device. Consequently, many *ad hoc* secured logic styles have been put forward. In the embedded security community, the so-called DPL (Dual-rail with Pre-charge Logic) family is overwhelmingly consensual. The DPL basically divide into two categories: "*power-constant*" and "*masked-power*" styles. The former requires careful backend stages, whereas the latter can be used by automatic CAD tools without any particular caution. The first category is comprised, amongst others, of SABL [452], WDDL [456], and anonymous logics [193, 55, 373]. For these gates to remain secure, the interconnect must be balanced too. To relieve this constraint, gate-level power masking has been introduced; two notable styles are RSL [441] and MDPL [359]. The idea behind these logics is to introduce a degree of freedom, enabling any of the dual rails to be used interchangeably. The extra degree of freedom is referred to as a random Boolean mask, and is supposed to be unknown to an attacker. The main handicap of masked logics is that they induce an overhead in terms of power consumption and design routability, and that they somehow delegate the security to a costly true random number generator. Additionally, the bit-level masking can, under some conditions (when the mask itself is leaked), be circumvented [455].

In this paper, we investigate the feasibility of implementing optimally secured unmasked logic. We argue that thwarting all known power attacks is possible (on netlist schematics.) The extra effort to provide is twofold:

1. most gates must be re-designed, because genuine standard cells are not secure enough. Notice that WDDL (based on standard cells) has been shown to open the door to vicious attacks based on accurate delay analyzes [439] that exhibit the early-evaluation weakness of CMOS standard cells;
2. the backend P&R flow must be secured, so as not to compromise the security of the logic gates.

Industrially speaking, these efforts are not deterrent because the consented investment is quickly amortized by high-volume productions. In low-volume products, FPGAs are generally preferred over ASICs. It is more difficult to map power-constant logics in reconfigurable devices, although some preliminary experiences have shown that using secured logics in FPGA is not utopian [125]. Additionally, robust-by-design FPGAs are showing up [222], which might foster, at medium term, reconfigurable commodities enabling constant-power reprogrammable designs.

The focus of this article is thus to provide the maximum security level, while remaining compatible with legacy design kits. There are indeed two strong incentives:

1. Provide a seamless integration in well established design flow, especially by making it possible for some standard gates to be reused verbatim.
2. Ease the integration of a secured design into a regular one. This operation is facilitated by the compatibility between the CAD flows.

The rest of the paper is organized as follows. The specifications of the secured logical gates design are dealt with in Sec. C.2. Then, the interconnect, supply and dummies

insertion issues are discussed in Sec. C.3. In Sec. C.4, the design methods presented in Sec. C.2 and C.3 are applied to the secured layout of a DES (NIST FIPS 46-3) co-processor. Finally, Sec. C.5 concludes the paper and provides some perspectives.

C.2 Secured Logic: SecLib

A design is projected into a set of logic elements by a synthesis step called "technological mapping". The logic elements in a cell library are in charge of the computation proper. The "SecLib" cell library, described in this section, is intended to be compatible, in terms of placement sites, with standard cells. This interoperability enables to reuse legacy cells for non-functional instances:

- **scannable flip-flops**, for synchronous designs pipelining and testability,
- **bufferization**, be it for capacitive load adaptation or skew balancing in clock tree generation,
- **clock-gating logic**, to freeze idle modules,
- **PN diodes**, for antenna effects correction,
- **filler cells**, for the N-well and power rails continuity in placement rows.

The constraint of placement-site compatibility with standard cells thus helps to reduce NRE (Non-Recurrent Engineering) costs. The DES co-processor illustrated in Sec. C.4 is comprised of mixed SecLib/standard cells.

SecLib, like other DPL libraries taylored for highly secured implementations, features security counter-measures at various levels: protocol, architecture, backend.

At the protocol level, a four-phase protocol enables to divide the computations into two steps:

1. the computation proper,
2. the precharge of the netlist.

The first step consists in the computation of one iteration, while the second re-initializes all the nets so that the circuit is ready to start a new computation afresh, for instance with all the nets in a same electrical state.

Additionally, most secured cells rely on a dual-rail encoding: every logical bit is in fact carried by two wires. Many representations exist; however, a common one consists simply in associating the value *false* (0) to a wire and the value *true* (1) to the other. The rationale is to make any transition on the two wires indiscernible. This fundamental hypothesis is trustworthy as long as two conditions are fulfilled:

1. Selecting one wire of the pair for probing or near-field antenna analysis [134] should be very difficult. This condition is guaranteed by the typical pitch. For instance, in a 130 nm process, two wires are separated by a pitch of 410 nm. At this scale, while not infeasible, the selection of one wire may be regarded as very difficult.
2. The two nets must be perfectly balanced and exhibit the same behavior in power consumption or propagation delay. This condition requires an *ad hoc* and *full-custom* design of the cells. All the possible sources of dissymmetry must be carefully analyzed and cured.

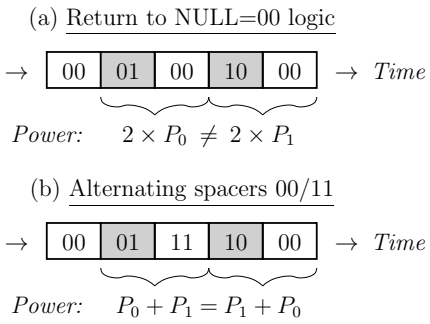


Figure C.1: Comparison of the (a) **return-to-null** and (b) **alternating spacer** logic, when the attack maximum voltage probe bandwidth is inferior than half the typical toggle frequency of the gates.

In dual-rail, every Boolean variable A is represented by a couple of two wires (A_0, A_1); when A is valid, $A = 0 \Leftrightarrow (A_0, A_1) = (1, 0)$ and $A = 1 \Leftrightarrow (A_0, A_1) = (0, 1)$. When A is not valid, $A_0 = A_1$. There exists two invalid states, namely: $A_0 = A_1 = 0$ and $A_0 = A_1 = 1$. Two DPL flavors can thus be used:

1. the **return-to-null logic** [317], where only $A_0 = A_1 = 0$ (or "NULL") is used and
2. the **alternating spacer logic** [420, 62], where both spacers 00 and 11 are used in an interleaved way: valid $\rightarrow$ 00 $\rightarrow$ valid $\rightarrow$ 11 $\rightarrow \ldots$

This signalization is also known as 1-out-of-2 delay-insensitive protocol, because any change in A_0 or A_1 carries an information (*evaluation* or *precharge* state.)

The "alternating spacer" signalization is more complex than the "return-to-null". However, it allows to better hide the data being manipulated if the attacker does not have access to an high-speed and/or high-bandwidth acquisition apparatus. Similarly, it can complement other counter-measures, such as random timing jitter addition to the signals: in such degraded experimental conditions, the actual acquisition quality is indeed lowered, thus impeding even a well-equipped attacker. In the case where the differential gates are not perfectly balanced (as in WDDL [456], where dual gates are dissimilar), the toggle of the *false* (0) output wire yields an average power dissipation P_0, whereas the toggle of the *true* (1) output wire dissipates $P_1 \neq P_0$. If the limited acquisition means of the attacker only allows him to access the power's average over two clock periods (or more), then:

1. the **return-to-null logic**, leaks $2 \times P_0$ (resp. $2 \times P_1$) when the gates evaluates to '0' (resp. '1'), whereas
2. the **alternating spacer logic**, leaks $P_0 + P_1 = P_1 + P_0$ for both evaluations, as shown in Fig. C.1. Consequently, the power side-channel does not reveal the Boolean value evaluated by the gate, although it is unbalanced.

The second type of logic is relevant for differential logics where the true and the false outputs are evaluated by gates that can be distinguished by a power leakage. The SecLib

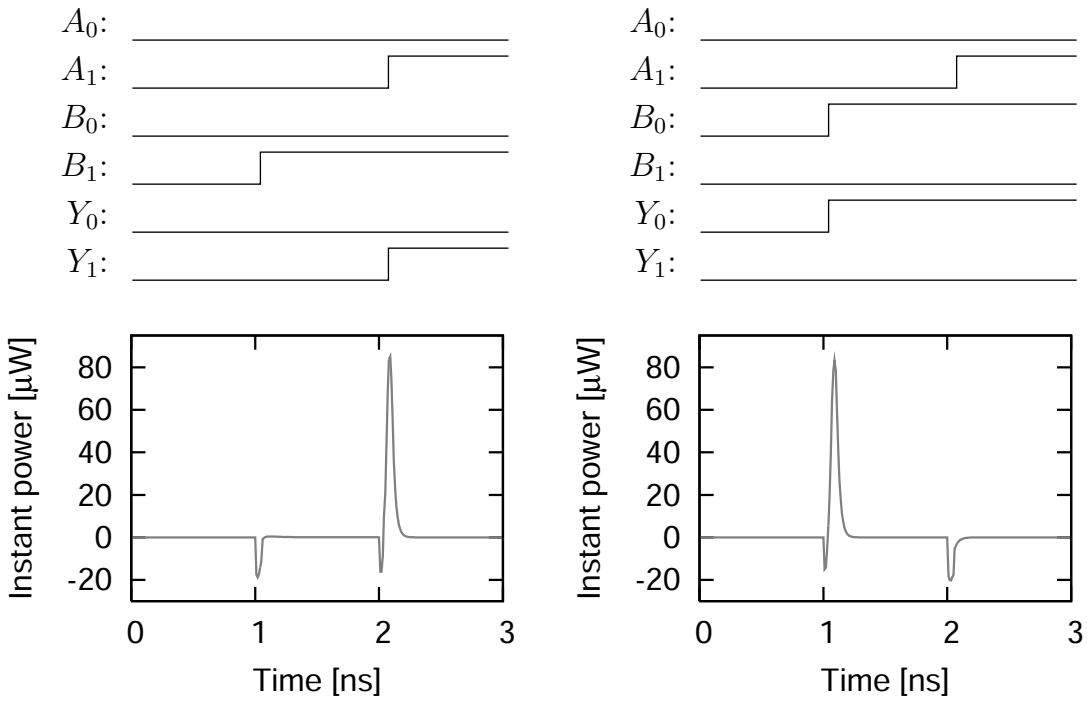


Figure C.2: Principle of the late/early power dissipation exploitable leak [439] illustrated on an unloaded WDDL `AND` gate.

cells presented below are designed for the two dual outputs to be indiscernible. This enables some optimizations in the gates transistor-level architecture.

The overall architecture of a representative SecLib gate is classical to the quasi-delay insensitive (QDI) logic [193]:

- the inputs synchronization disables anticipated evaluation. The gate timing is thus unconditional to the data. This feature protects the gate against the signature differences of unsynchronized DPL caused by variations of input delay time. The SPICE (Simulation Program with Integrated Circuit Emphasis) simulation of this behavior is illustrated in Fig. C.2.
- the inputs configuration decoding $(A, B) \mapsto (C_{00}, C_{01}, C_{10}, C_{11})$ is well suited for an indiscernible processing. Notice that, for unbalanced functions, the computation part is forced to be symmetric by the use of dummy gates (*cf.* Fig. B.1 schematic on the *right*.)

The schematic of a QDI secured `AND` gate is given in Fig. B.1. Notice that QDI cells are suitable for both synchronous or asynchronous circuits. The transistor-level architecture of the C-Elements [414] and of the computation logic is detailed below. A so-called "symmetric" architecture is chosen for the C-Elements: in this architecture, the two inputs $(A_i, B_j), (i, j) \in \{0, 1\}^2$ that are rendez-vous'ed are indiscernible. Moreover, some transistors are added to fix the potential nets that would otherwise leak information

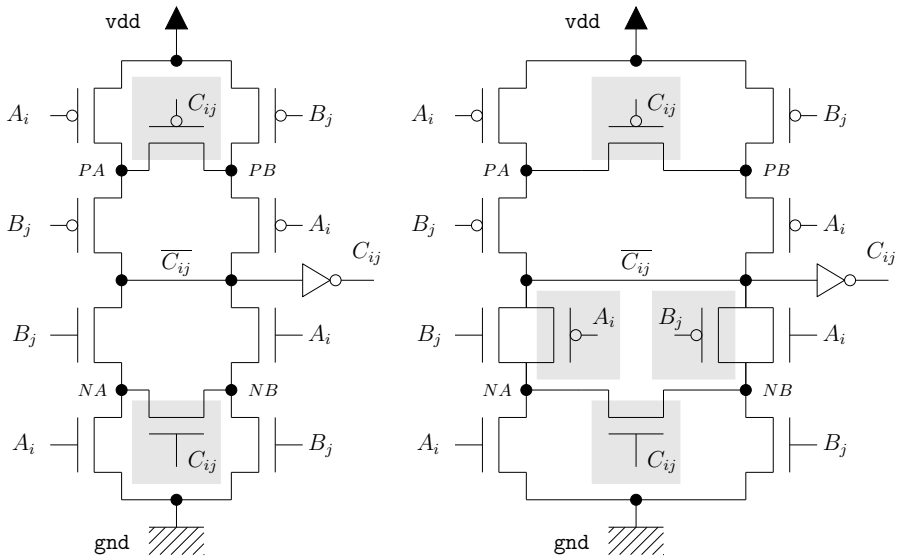


Figure C.3: Regular (*left*) and secured (*right*) C-Element transistor-level netlist.

through parasitic capacitive memorization of the previous state. The actual number of such transistors is limited to the nets that are not reset automatically by the return-to-null protocol.

The C-Element netlist is represented in Fig. C.3. Driving transistors are depicted normally, whereas keeping transistors are shaded. Notice that at the layout level, *driving* transistors must have a larger aspect ratio W/L (gate width/length) than their *keeping* counterparts.

Functional gates with unbalanced truth tables (such as the `AND`), requires special attention. We further discuss the implementation proposed in [193], where dummy `OR`s are added in the computation part of the gate. The `3OR` netlists can be optimized in terms of `NAND`s and `NOR`s (see Fig. B.1), as this is commonly done in CMOS logic (where gates are natively inversing.)

C.3 Secure Routing: Shielded DRC-clean Backend-Duplication

C.3.1 Routing Objectives

This section presents a secured routing methodology suitable for differential netlists. Some techniques have already been described in the literature, such as the "fat wires" or the "backend duplication". However, they lack to encompass some constraints related to various domains:

– **security**: cross-talk represents a vulnerability. Differential pairs must be protected against this phenomenon. It becomes increasingly disquieting as metallization pitches shrink.

– **manufacturability**: for a circuit to be accepted in foundry, some design rules must be satisfied. A design rule checker (DRC) can verify that none of them are violated. The minimum *spacing*, *width*, *etc.* are local rules that can be checked during the routing step. The *slot* and *density* rules are global constraints to be verified on a wide layout area. When minimally sized wires are used, there are no risk to create large areas of metal without holes: as a consequence, slot rules are not a concern. The density of every metal layers must be bounded, typically within the range $[20\%, 80\%]$; usual designs often have low-density areas, and thus do not verify this constraint naturally. For this reason, the design flow must explicitly take it into account.
– **power supply**: some of the routing resources, customarily called "stripes", must be devoted to convey current flows from `vdd` and to `gnd`.

C.3.2 Routing Strategy

In order to meet all the routing objectifs, we use the following constraints:
– One track over two is reserved for the dual net routing.
– Another track over two is reserved for the shield against potentially aggressive neighboring nets.

As explained in [191], the reservation can be achieved in a straightforward way by instantiating routing constraints in the P&R tool. Under Cadence SOC/Encounter®, the constraints can be expressed in TCL (Tool Command Language) as:

```
createRouteBlk  9.945 0 10.145 510.86 4 -name M4_1
createRouteBlk 10.355 0 10.555 510.86 4 -name M4_2
createRouteBlk 10.765 0 10.965 510.86 4 -name M4_3
# Track available to route one 'true' signal
createRouteBlk 11.585 0 11.785 510.86 4 -name M4_5
createRouteBlk 11.995 0 12.195 510.86 4 -name M4_6
createRouteBlk 12.405 0 12.605 510.86 4 -name M4_7
# Track available to route another 'true' signal
createRouteBlk 13.225 0 13.425 510.86 4 -name M4_9
# Et caetera ...
```

Afterwards, once set, they can be visualized from the floorplan viewer, using the configuration panel of Fig. C.4.

Avoiding cross-talk can be achieved by two means: either shielding with a global net or spacing. The shielding solution is the preferred one because the spacing solution has a major drawback: it risks to violate the DRC rule that requires that the metal density be greater than a lower bound, even if the enforcement of this rule is indeed negotiable with the manufacturer. The risks of a poor density materialize in the unevenness of metal wires height because the CMP (Chemical-Mechanical Polishing) manufacturing process operates on heterogeneous surfaces. The non-respect of the minimum density rule might cause both a yield and a security issue; the security issue arises from the vertical discrepancies to the otherwise horizontally balanced pairs.

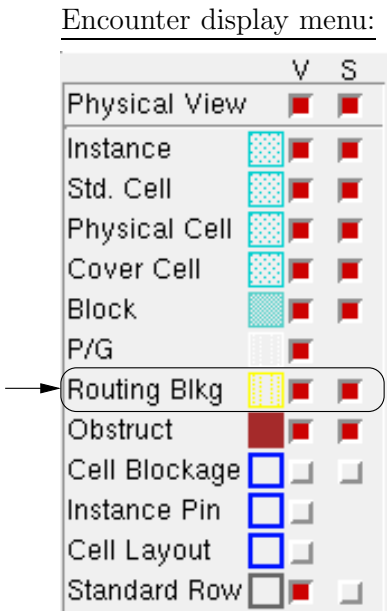


Figure C.4: Controls to display the routing blockages.

Consequently, shielding with a metal net is mandatory. In a view to avoid wasting routing resources, the shielding nets can also chosen to be the supplies `gnd`/`vdd`. Thus, half of the routing resources are dedicated to both shield data nets and route the supplies down to the cells. The figure C.5 shows the allocation of the routing tracks for two layers of metal. The free tracks, represented by dashed lines, are available to route differential pairs. The power planning is thus obtained by the horizontal and vertical translation of the elementary $(4 \times \texttt{PITCH}) \times (4 \times \texttt{PITCH})$ cell represented shaded in Fig. C.5.

The highly interconnected power network ensures that the voltage levels remain stable. By itself, the power network suffices to fulfill the minimum density design rule. In

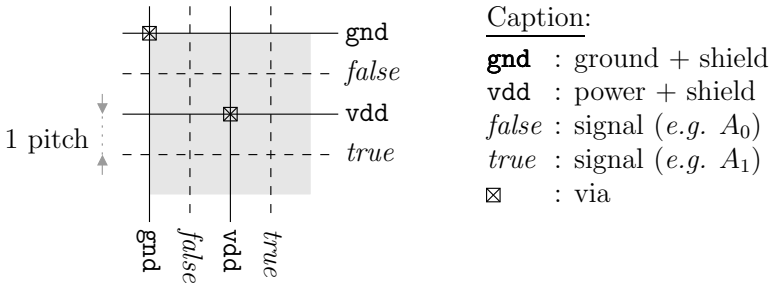


Figure C.5: Routing tracks allocation on two consecutive metal layers using the shielded DRC-clean backend-duplication routing method.

a 130 nm process, where the routing width and pitch are respectively 200 and 410 nm, the density is indeed $200/(2 \times 410) = 24\% > 20\%$. Notice that in congested regions, the maximum density complies with the DRC: $(2 \times 200)/(2 \times 410) = 49\% < 80\%$. To improve the CMP quality, the density should ideally be uniform. Metal dummies can therefore by added everywhere the routing tracks are not filled by data wires. In the backend duplication method, dummies rectangles are duplicated at the same time as the data wires. Technically speaking, a translation is performed on all the elements of the following DEF sections: `COMPONENTS`, `PINS`, `BLOCKAGES`, `SPECIALNETS`, `NETS`, `FILLS`. Figure C.6 shows one portion of a layout with the connections to the ground & power rings before and after duplication.

The routing of data signals is thus realized in a regular mesh of interleaved ground and power nets. The immunity to noise of the data signal is thus optimal. In addition, the dense power supply mesh reduces the local voltage variations at the gate-level, otherwise caused by voltage drops along long power lines. This positive side benefit enhances both the yield and the security of the design. The dense routing of the shielding mesh also hides the underlying logical cells, which makes their visual recognition very hard. Thus, this shield protection method also complicates the reverse-engineering (with tools such as automatic gates recognition softwares [408]). In addition, it prevents micro-probing, since probing needle can hardly pass through the remaining holes (only 51% of the space).

Notice that metal dummies are difficult to insert in the lower metallization levels of the cells. We found it convenient to design the cells in such a way that the density constraints are met by design in the metal layers used by the cell (typically M1 and M2.) The metal filling is indeed fully compatible with the "cage" strategy presented in Sec. C.2.

C.4 DES Datapath Case-Study

C.4.1 Performances Evaluation

As a case-study, a DES co-processor [195] has been designed in order to assess the performances of above-mentioned methodology. The performances in terms of area are given in Tab. C.1. For the secured DES, the number of instances (#inst.) and of unique instances (#!inst.) are given as the sum of standard cells and of full-custom SecLib cells. Both instances have been validated functionally by digital simulation, and are clean from errors regarding STA (Static Timing Analysis), DRC (including antenna rules) and LVS (Layout Versus Schematic).

The metal use of the proposed methodology is summarized in Tab. C.2.

The insertion of the `gnd`/`vdd` mesh does not impede the balancing between the dual-rail pairs capacitances. The parasitics are extracted from the placed-and-routed design, using a database of capacitances precharacterized with a 3D field solver. The capacitance of the 2 610 dual-rail nets is computed. The average statistics are given in Tab. C.3. As expected, the extractor finds more capacitances in the design with the `gnd`/`vdd` stripes mesh. In both cases, the main contribution for the capacitances is the interconnect.

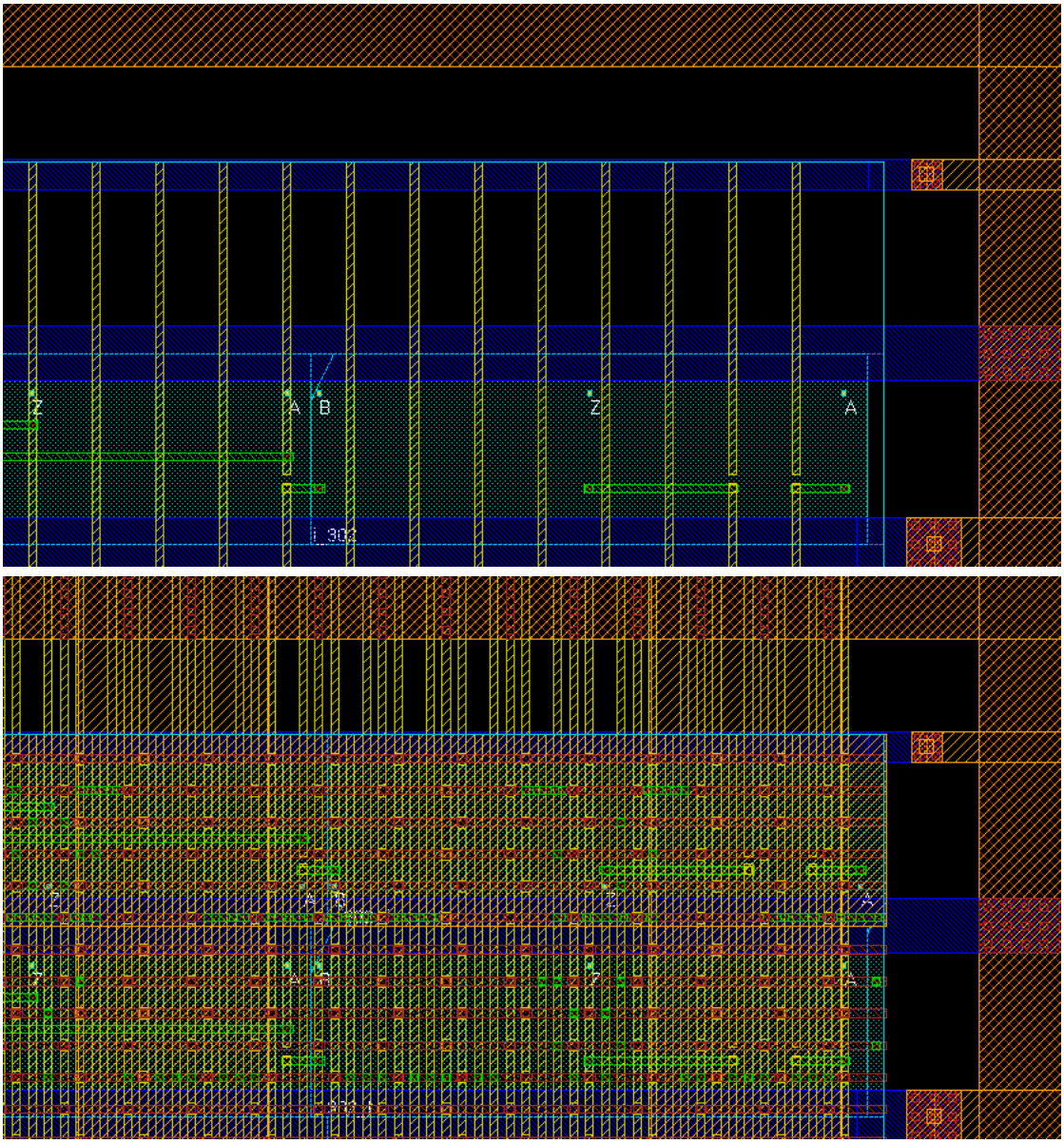


Figure C.6: Upper right corner of the DES datapath layout, before (*top*) and after (*bottom*) duplication.

Table C.1: Area comparison between the regular and the secured DES datapaths.

Regular DES (standard cells)			
#inst.	#!inst.	Area	Density
1 497	68	25 368 μm^2	95 %

Secured DES (SecLib)			
#inst.	#!inst.	Area	Density
862 + 2 295	7 + 9	382 871 μm^2	95 %

Table C.2: Metal layers preferred direction (Horizontal or Vertical) and assigned purpose(s.)

#	**H/V**	**Purpose**
1	H&V	Internal SecLib interconnect (inside C-elements, `3OR`s, *etc.*)
2	H&V	SecLib interconnect (C-elements between them, *etc.*) and pins + neighbor cells direct connection of face-to-face pins
3	H	Suited to maximize pin accessibility
4	V	To connect pairs of placement rows
5	H	Spare horizontal routing
6	N/A	Unusable because of larger pitch

Table C.3: Cumulated capacitances statistics of the dual-rail placed-and-routed SecLib DES netlist without and with stripes.

	No stripes	**With stripes**
Wire	70 pF	121 pF
Gate	55 pF	55 pF
Total	125 pF	176 pF
Wire/Total	56 %	68 %

One relevant "static" parameter to evaluate the security of a DRL design is the dispersion of the ratios C(true)/C(false) for all the dual-rail pairs. This quantity can be greater or smaller than one, depending on the unbalancedness direction. In order not to favor one of the wires in a pair, the Neperian logarithm of the ratio is considered instead. The histogram for these quantities is given in Fig. C.7. The standard deviation is equal to:

- 1.53×10^{-3} for the design without stripes and
- 0.48×10^{-3} for the design with stripes.

It must be noted that these values are extremely low. In this case, the limited series $\log(1+\epsilon) = \epsilon + \mathcal{O}(\epsilon^2)$ applies. The discrepancy is thus roughly equal to 1.5 ‰ (resp. 0.5 ‰) without stripes (resp. with stripes.)

Finally, for the results to be perfectly clear, it must be underlined that the extracted deviations are caused by dual-rail pairs cross-coupling only (despite a systematic shielding strategy); in particular, the technological dispersions are not taken into account in this study. Nonetheless, this ineluctable phenomenon is expected to take on more and more importance as the routing pitch reduces.

C.4.2 Comparison with Related Works

One previous work, by Kris Tiri [460, 225], presented a comprehensive design flow taylored for the WDDL logic. However, this logic has been shown to feature a weakness [439] against which SecLib resists. Independently of this issue, it is interesting to compare the performance of a DES datapath co-processor implemented in SecLib and in WDDL from the same VHDL source code. A system-on-chip (SoC), called SecMat V3 – inheriting SecMat V1 architecture, has been designed to compare the security of a DES module implemented in SecLib and in WDDL [456]. The floorplan is depicted in Fig. C.8.

The table C.4 reports the minimal area required to place and route the WDDL DES datapath. Apart from seven standard cells used as such, two instances of reshaped gates are required. They are the unitary `AND` and `OR` gates from the standard library, whose interface has been adapted to be symmetric. A total of 6 060 such gates are needed after logical synthesis, corresponding to 3 030 logical `AND` (couple {true = `AND`, false = `OR`}) and logical `OR` (couple {true = `OR`, false = `AND`}).

It appears that the WDDL module fails to be routed with a 95 % density. As shown in Tab. C.4, the best density is 35 %. This clearly shows that the WDDL design using the routing methodology presented in Sec. C.3 is in the *wire-domain*, as opposed to the *logic-domain*.

In [225], a working WDDL circuit is shown to be successfully designed with only a factor 3.1 of area increase over a standard CMOS implementation. However, the routing shielding is not discussed in [225]. The area increase presented in Tab. C.4 can also be reinterpreted as:

- a 4.4 times area overhead for the logic, placed at 95 % density, composed to
- a 2.7 (=95 %/35 %) times area overhead for the differential routing shielding.

(Without stripes) (With stripes)

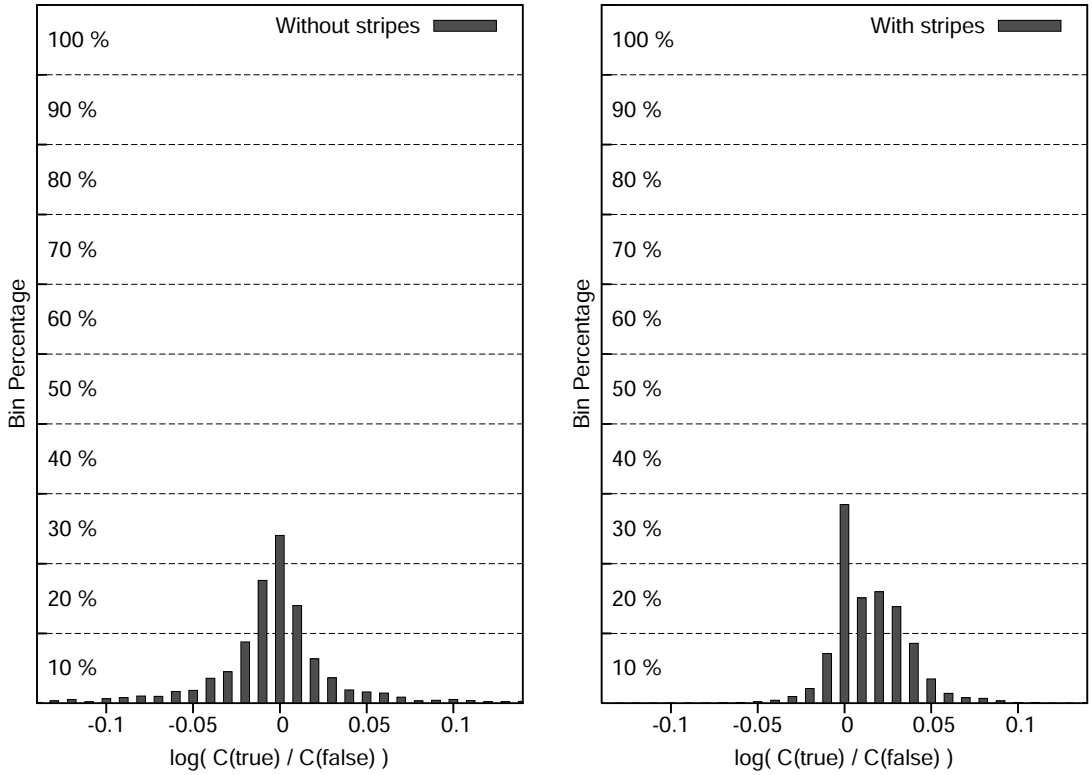


Figure C.7: Distribution of the unbalancedness of dual-rail pairs capacitances in DES SecLib.

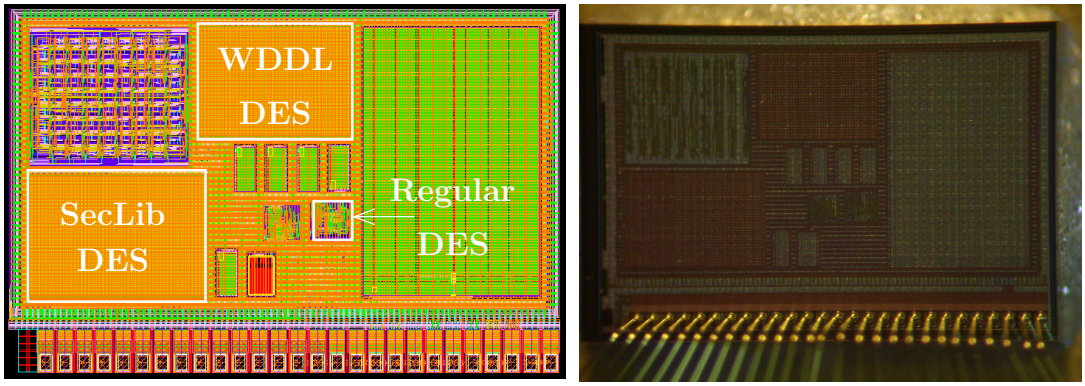


Figure C.8: Floorplan of the SecMat V3 SoC (*left*, from the Cadence Virtuoso CAD tool renderer – *right*, from a photographic picture).

Table C.4: Area figures for the WDDL DES datapath.

Secured DES (WDDL)			
#inst.	#!inst.	Area	Density
404 + 6 060	7 + 2	299 824 μm^2	35 %

Table C.5: Cumulated capacitance statistics of the dual-rail placed-and-routed WDDL DES netlist without and with stripes.

	No stripes	**With stripes**
Wire	57 pF	95 pF
Gate	25 pF	25 pF
Total	82 pF	120 pF
Wire/Total	68 %	78 %

The fact that the logic area overhead is greater in our case than in [225] (4.4 *versus* 3.1) can be explained by the fact that we have not tried to optimize them, since the design is in the *wire-domain*, not in the *logic-domain*. For instance:

- no complex gates (`AOI`) are used, only plain `AND` / `OR`,
- the WDDL register is made up of four regular `DFF`s, instead of two `DFF`s and two `NOR`s.

The statistics about the capacitances of the 3 684 dual-rail nets are represented in Tab. C.5 and in Fig. C.9. The standard deviation of the C(true)/C(false) ratio is equal to:

- 1.96×10^{-3} for the design without stripes and
- 0.58×10^{-3} for the design with stripes.

These values are slightly higher than the ones obtained from SecLib because the "`AND`" and "`OR`" standard cells input capacitances (though they do not represent the main part of the total capacitance) are not perfectly equal:

- `AND:A` is 1.99 fF and `AND:B` is 1.75 fF, whereas
- `OR:A` is 1.84 fF and `OR:B` is 1.60 fF.

The three DES co-processors (standard cell, SecLib) have been optimized for area, not for speed. They have been synthesized with the constraint to remain functional with a clock period of 15 ns. Indeed, the critical path is another part of the SecMat V3 SoC, namely between the micro-processor and the 32 kbytes memory. At 66.7 MHz, each of the three DES instances can process encryptions and/or decryptions at [195]:

- 266.7 Mbit/s $= \frac{64 \text{ bit}}{16 \text{ clock}} \Big/ \frac{15 \times 10^{-9} \text{ s}}{\text{clock}}$
  in DES-CBC with a 56-bit key, or

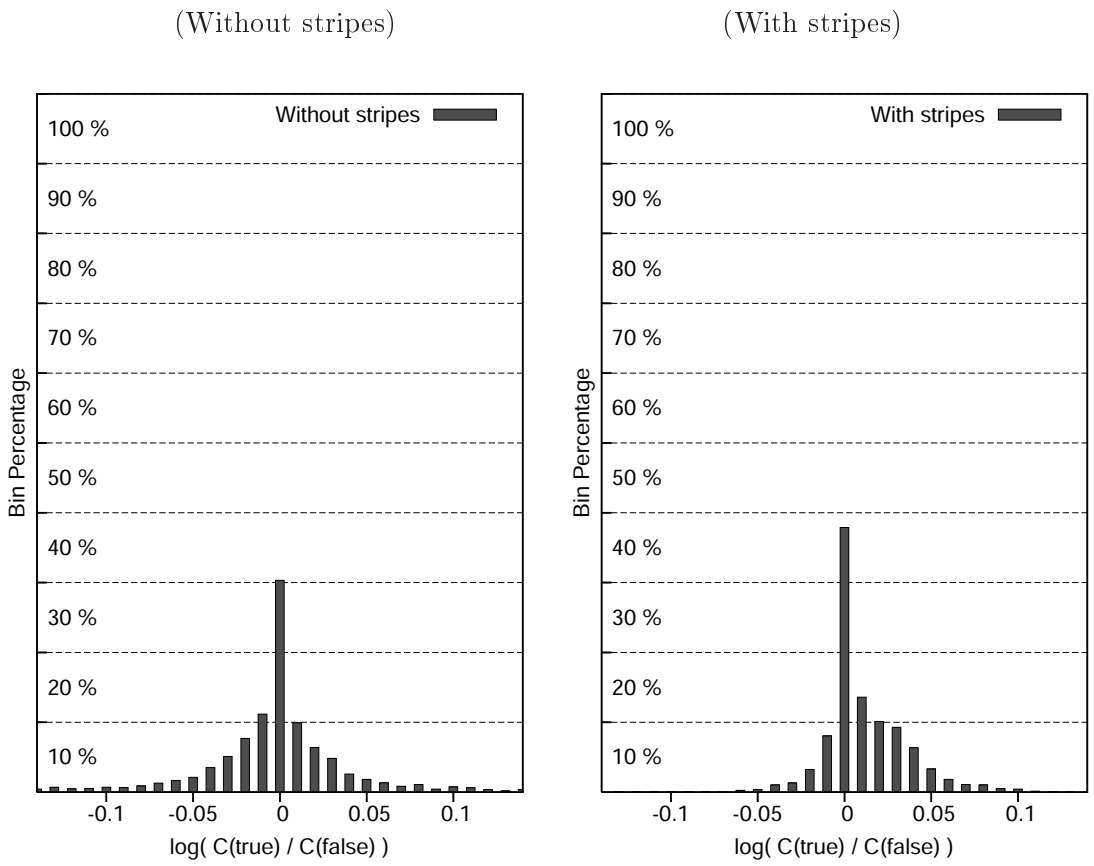


Figure C.9: Distribution of the unbalancedness of dual-rail pairs capacitances in DES WDDL.

– 88.9 Mbit/s $= \frac{64 \text{ bit}}{16\times3 \text{ clock}} \Big/ \frac{15\times10^{-9} \text{ s}}{\text{clock}}$
in 3DES-CBC with a 112-bit key.

C.5 Conclusion

This paper revisits the design of statically secured cells suitable for constant-power custom cryptographic ICs. Most previously proposed gates are vulnerable to a power attack exploiting the inputs skew. Therefore, this article focuses on a logic style, referred to as SecLib, in which gates inputs are systematically resynchronized. We emphasize the topological issues raised by the symmetric routing constraints. The question of the positions of the pins is extensively discussed. Then a method to achieve a DRC-clean and optimally secured (parallel and shielded) routing is presented. The shield between dual-rail pairs (necessary to properly avoid cross-talk, fatal to the security of the interconnect network) is also used to convey the ground and power global signals pervasively to every standard cells, thus guaranteeing a perfect stability in the energy delivery. The paper concludes positively on the feasibility of industrial-strength secured cells libraries. The realization of a constant-power DES cryptoprocessor with legacy CAD tools in a 130 nm technology proves that non-masked security at the gate-level with balanced and shielded routing can be implemented in practice.

Our main contribution is to show that when the maximal effort is spent on both the logic and the routing, then the interconnect is the limiting factor as for the implementation area. Let apart the cost overhead, the previously proposed secure backend methodologies (for instance "SecLib" logic and "backend duplication" constrained P&R) can thus become industrial commodities. Based on this proof-of-concept, taylored optimizations, typically based on security *versus* cost trade-offs, can be thought of. There is certainly a large margin for drastic improvements in terms of adequation between a given security model and a required protection profile.

C.6 Acknowledgements

The work presented in this article has been partly funded by the Conseil Régional de Provence-Alpes-Côte d'Azur (PACA) and the French National Agency for Research (ANR) through the MARS (ACI SI 2004) grant.

The authors thank the AST division of STMicroelectronics Rousset (France) for its support in the SecMat (Sécurité du Matériel) project. In addition, the authors are grateful to Ronan Keryell (GET / ENSTBr, "Trusted Computing Platform" project) for his valuable advices and encouragements.

Appendix D

Security Evaluation of WDDL and SecLib Countermeasures against Power Attacks

Extended version of article [175]

Authors: *Sylvain Guilley, Laurent Sauvage, Philippe Hoogvorst, Renaud Pacalet and Guido Marco Bertoni*

Abstract

Logic styles with constant power consumption are promising solutions to counteract side-channel attacks on sensitive cryptographic devices. Recently, one vulnerability has been identified in a standard-cell based power-constant logic called WDDL. Another logic, nicknamed SecLib, is considered and does not present the flaw of WDDL. In this paper, we evaluate the security level of WDDL and SecLib. The methodology consists in embedding in a dedicated circuit one unprotected DES co-processor along with two others, implemented in WDDL and in SecLib. One essential part of this article is to describe the conception of the cryptographic ASIC, devised to foster side-channel cryptanalyses, in a view to model the strongest possible attacker. The same analyses are carried out successively on the three DES modules. We conclude that, provided that the backend of the WDDL module is carefully designed, its vulnerability cannot be exploited by the state-of-the-art attacks. Similarly, the SecLib DES module resists all assaults. However, using a principal component analysis, we show that WDDL is more vulnerable than SecLib. The statistical dispersion of WDDL, that reflects the correlation between the secrets and the power dissipation, is proved to be an order of magnitude higher than that of SecLib.

Keywords: side-channel attacks, differential power analysis, secured logic style, WDDL, SecLib, backend-level countermeasures.

D.1 Introduction

Much equipments must conceal secret information, such as personal data, credentials or intellectual properties. Now, these devices can be stolen or simply bought by any attacker who wishes to retrieve the secrets. Indeed, attackers can eavesdrop the information directly within the equipment. In this context, the digital information can no longer be protected by sole cryptographic means. For this reason, many applications delegate the low-level security to a specialized circuit. It usually takes the form of a smartcard, a trusted platform module (TPM) or an embedded crypto-processor. For instance, in some countries, the access to operated mobile telecommunication networks is protected by a subscriber identity module (SIM) card. The authentication at automated teller machines (ATMs) is often realized by a smart card. Worldwide, personal computers are equipped with TPMs. Some FPGA manufacturers now implement on-chip configuration bitstream decryption.

To avoid on-board bus probing, the secured system consists most of the time of a monolithic ASIC. Securing those chips is of major importance. Two threats have been identified in the last decade: side-channel attacks and fault injection attacks. The principle of fault attacks is to force the circuit to malfunction so as to gain illegitimate information [144]. These attacks are very powerful and some circuits have been successfully broken with this technique. However, given that this attack is active, the circuit can embed fault detection logic. If an error is detected, the circuit can for instance erase its secrets, which implies that an attack might require to sacrifice many circuits. Side-channel attacks consist in observing whatever physical emanation that leaks from the circuit, in a view to derive some secret information about the secrets it handles. They are more sneaky because they are passive: if they are carried out carefully, the circuit is not aware that it is being attacked. Usual side-channels are the timing, power consumption [290] or electromagnetic emanations.

Many successful attacks on unprotected circuits have been reported publicly since 1996. Standard side-channel attacks (SCAs) are SPA [249], DPA [249, 308], inferential power analysis (IPA) [121], CPA [60, 258], EMA [134, 353] and template attacks [69, 376, 13]. To mitigate side-channel attacks, several types of countermeasures have been proposed and implemented. It is possible to balance or randomize the sensitive design at the algorithmic, logical or physical levels: the overall strength of the design will be that of its weakest countermeasure. The security evaluation of the protected circuits usually proves that the efforts to spend to break the circuit is higher than without protections. Unfortunately, many protected implementations were actually partially broken, albeit with more expansive means. Two reasons are mentioned to explain the attack success. Either the attacker exploits a leakage that is not covered by the countermeasure. Or the hypothesis about a countermeasure is made at one level, say logical, but is not ported at a lower level, say physical.

It is now widely admitted by the side-channel community that the SCAs have the potential to extract information about any net of the design. It is thus very often advised to protect the circuit down to the logic gate. In the field of gate-level countermeasures,

two options are generally considered: static or dynamic countermeasures. The goal of the former is to ensure a power-constant execution, whereas the second consists in ensuring a power-constant execution in average, with the help of an ancillary TRNG.

In this article, we specify an attacker that is able to perform DPA, CPA and template attacks. We investigate experimentally her potential to break implementations protected against the specified attacks, with both logical and physical countermeasures. More specifically, the WDDL logic [456] with wire shielding is assessed. We observe that a reported flaw against WDDL [439] cannot be exploited. Additionally, we investigate another logic, called SecLib [193], immune from the WDDL flaw. The second goal of the paper is to quantify the security gain when switching from WDDL to SecLib. This information is very valuable to adapt the security level to the cost of the assets to protect.

The rest of this article is structured as follows. The ASIC designed for the security evaluation is described in Sec. D.2. The three DES modules implementation is detailed in Sec. D.3. In Sec. D.4, the attack methodology and results are given. Finally, section D.5 concludes the paper and opens further research perspectives.

D.2 Prototype ASIC Dedicated to Side-Channel Information Leakage Evaluation

A dedicated ASIC has been designed to evaluate the security level reached by the two competing logic styles. In the following, we refer to this chip as "SecMat v3". SecMat v3 has been taped-out on 2007 January 3rd (STM 0.13 μm technology HCMOS9GP with 6 layers of metallization) through the CMP (Circuits Multi-Projets) silicon broker [88]. The ASIC's die area is 4.4 mm^2 and contains 2.4 million transistors. The circuit is DRC & LVS clean and has been tested fully functional. A picture of the floorplan and of the acquisition printed circuit board (PCB) is given in Fig. D.1. The knowledge of the accurate RTL description of the system is an important feature: it enables us to relate side-channel analyses to the circuit's operations.

The architectural choices made during the design of SecMat v3 are detailed in this section.

D.2.1 Security Evaluation Target: ASIC *versus* FPGA

It makes sense to attack both targets. However, in our context, we endeavor to:

1. implement sound and robust countermeasures and
2. foster the access to the side-channel. Indeed, to increase our level of confidence in an evaluation, the usual methodology consists in choosing the experimental setup that maximizes the attack's strength.

The ASICs are thus compared to the FPGAs in these two respects.

The implementation of some countermeasures is either impossible or more difficult in FPGAs. Full-custom logic styles cannot be implemented in FPGAs, since the finest reconfiguration grain is the look-up table (LuT), and not the transistor as in ASICs. The

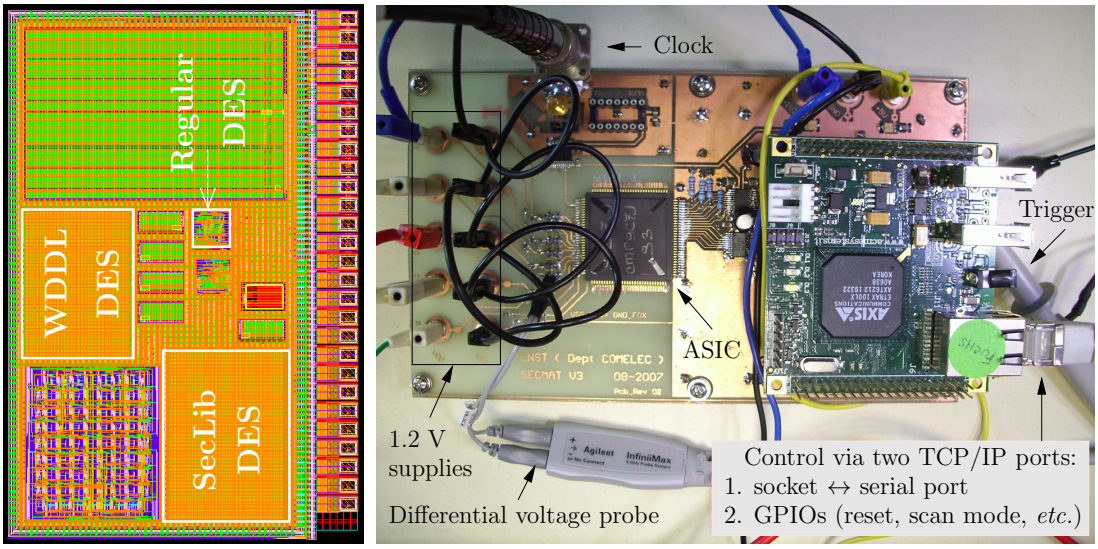


Figure D.1: Floorplan and acquisition board of the SecMat v3 ASIC.

placement can be constrained in both targets. Kris Tiri showed how to place WDDL in `F` and `G` LuTs in Xilinx FPGAs [459, 458]. Altera proposes the `logiclock` feature to achieve a similar result, albeit at the logic array block (LAB) level. Native FPGA CAD tools do not implement pair-wise dual-rail routing. Concerning ASICs, some tools start to feature this functionality. For instance, Cadence "chip optimizer" provides a space-based router. Although this post-processing functionality is intended to balance only "special wire" couples, it is conceivable to declare all the nets to be special. Nevertheless, other strategies to achieve this functionality have emerged: fat-wire routing [457] and backend duplication [191] operate on top of the CAD tool. In FPGAs, these methods would require the knowledge of the interconnect resources and the ability to forge a bitstream. Additionally, either the routing graph description must be changeable (in the fat-wire method, the channel width must be halved), or it must be possible for the user to set constraints (in the backend duplication, every other routing track must be blocked). The shielding of signals seams difficult in FPGAs: there are no publicly available papers dealing with this aspect.

Finally, the accurate power measurement in FPGAs is a challenge: spying a part of the FPGA consumption is possible under some product families. However, this constrains the module under test to be placed in a partition of the floorplan close to the power pads. Nonetheless, no application note guarantees that the power will not be modulated by the neighbor logic. As too many parameters remain unknown in FPGA designs, we opted for an evaluation in an ASIC, where every aspect is under control.

D.2.2 System-Level Architecture

The DES modules must be as indiscernible as possible. Hence the choice to place them on a same silicon die.

Besides, SecMat v3 is a system on chip (SoC), where the modules are slaves of a CPU, playing the role of the master. The interconnect is based on the VCI (Virtual Component Interface [8]) standard. Seen from the CPU, the DES modules share the same interface, and differ only from their addressing space. This organization greatly facilitates their control: the same program is typically used for all DES modules. This program repeatedly installs the cryptographic data (key and message) in each module's memory, asserts a line to trigger an oscilloscope and launches the encryption.

As the main goal of the ASIC is to realize accurate and fair side-channel measurements, a couple of power pads, called (`gnd_des`, `vdd_des`) is devoted specifically to the DES modules energy supply.

Another power requirement is to avoid coupling between the DES modules and other parts of the ASIC (CPU, pads, *etc.*) The solution to lower the "substrate noise" is to insulate the ground of the DES modules from the wafer bulk. The HCMOS9GP technology is triple-well: the `NISO` CAD layer allows to vertically insulate the P-well of a region. The addition of the `NISO` mask cannot be done by automatic placers and routers, such as Cadence SOC/Encounter. Therefore, we wrote a SKILL script that post-processes the layout by adding a surrounding `NISO` rectangle around every DES module. The same script also computes the equivalent diode created between `gnd_des` and the bulk; this information is indeed required by the LVS tool.

Anyway, it remains essential to avoid I/O pad activity during the encryption, especially if the pads carry sensitive data (such as a key). In all the experiments presented in the remainder of this paper, there are no I/O operations during the cryptographic operations.

As already stated, the goal of the ASIC is to be able to measure as accurately as possible the power dissipation of the DES modules, with the additional constraint that the power measurement be the same for all the modules. We opted for a shared power supply for the three blocks, but distinct from that of the rest of the core. The modules can be disabled by clock gating, so that only the attacked module absorbs energy. The clock gating suppresses the dynamic power consumption but not the static leakage current. However, given that the deactivated modules are left in a random state, no relevant information is expected to be leaked this way. For the sake of completeness, we mention that a constant leakage of 180 μA is measured on SecMat v3. In Fig. D.11 at page 136, a 9 mV offset is observed through a 50 Ω "spy" surface-mounted component (SMC) resistor.

A module, called "power management", decides whether the clock delivered to the DES modules is active or zeroed. The architecture of the "SecMat v3" SoC with the clock gating controller is depicted in Fig. D.2.

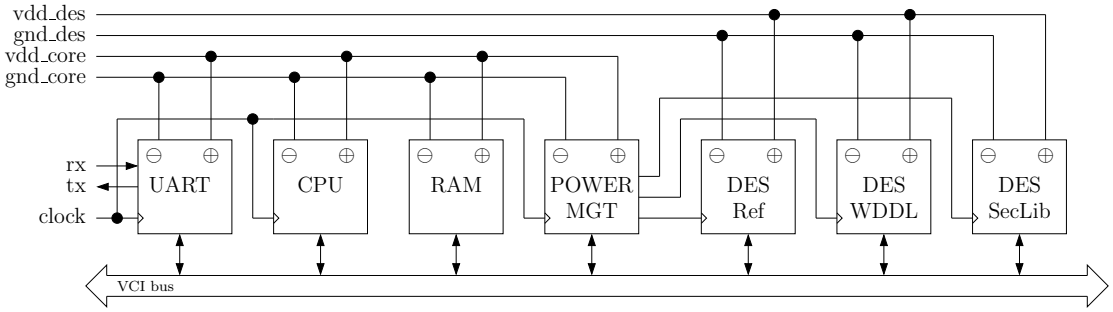


Figure D.2: SecMat v3 system-level power management.

D.3 Reference, WDDL & SecLib DES Modules

The data encryption standard (DES [336]) was chosen as the algorithm to evaluate the security of WDDL and SecLib countermeasures. This algorithm is the preferred one in ASIC implementations, because it is very small, and because of the confidence people have on its cryptographic strength (when used as triple-DES with three distinct keys). For example, DES is used in the electronic passport, in Europay-Mastercard-Visa (EMV) banking applications and in the bitstream encryption for Virtex 2 Xilinx FPGAs.

The architecture of the DES co-processors of SecMat v3 is detailed in [195]. It is an iterative implementation that processes 64-bits of data and that schedules the round key in parallel with the data encryption; one round of DES is thus computed each clock period. In our setup, the DES is made to operate on one single message block, according to the following schedule:

- clock period 0–7: byte-wise key loading from RAM,
- clock period 8–15: byte-wise message loading from RAM,
- clock period 16–31: encryption (16 rounds), in dedicated registers,
- clock period 32–39: byte-wise ciphertext saving into RAM.

For a fair comparison, the modules were designed to be as similar as possible. The VHDL source code is shared. The reference module has been realized using unprotected gates and straightforward automatic CAD tools. More precisely, we have used the Cadence toolchain for the design (`bgx_shell` for the logic synthesis, `SOC/Encounter` for the place/route step and `icfb` for the layout finishing) and Mentor Graphics `calibre` for the verifications (DRC and LVS). The WDDL and SecLib modules resort to advanced physical design techniques, that differ only regarding their logic style.

A description of the three DES modules embedded in SecMat v3 is already provided in [188]. We summarize the main security attributes of these modules in this section.

D.3.1 Logic Styles

Constant-power computations often use a *dual-rail with precharge logic* (DPL). This logic is also known as *dynamic differential logic*. The protocol of this logic consists

of two phases: precharge and evaluation. The precharge phase allows to start new computations from a known electrical state. It thus prevents unexpected transitions between two computation steps. The dual-rail signalization of the data is conveyed by two wires for each Boolean variable: $\texttt{NULL} = 00$ while in precharge and $\texttt{VALID} \in \{01, 10\}$ while in evaluation. Therefore, every evaluation consists in the transition of exactly one wire ($00 \rightarrow 01$ or $00 \rightarrow 10$). If the design is adequately balanced, which transition occurred is indiscernible by an attacker.

D.3.1.1 State-of-the-art about DPL.

In 2002, Kris Tiri introduces the "Sense Amplifier Based Logic" (SABL) logic style [452], which aim is to make power consumption independent of both the logic values and the sequence of the data. It is therefore the first DPL proposal. Its principle consists in combining Differential and Dynamic Logic (DDL) like in the "Dynamic Cascode Voltage Switch Logic" (DCVSL) style, while fixing second order asymmetry in the gate (especially for complex logic functions), due to parasitic capacitances [371]. This allows to decorrelate the power consumption from the inputs. In 2006, Marco Bucci *et al.* [61] show that the balance of DPL gates can be improved by adding a systematic discharge after the evaluation. The resulting computations are thus based on a ternary pace: (1) pre-charge, (2) evaluation and (3) post-discharge. When applied to SABL, simulations reveal that a gain of two-order of magnitude is obtained in terms of balance.

As these techniques require the full-custom design of new standard cells, Tiri proposes two years later the "Wave Dynamic Differential Logic" (WDDL) style [456]. WDDL uses a standard cell flow, where an original single-ended gate netlist is duplicated to obtain a differential netlist. In addition, the precharge is not global; instead precharge values are imposed only at the inputs, and propagate as a "wave" through the combinatorial netlist. Finally, the total load capacitance is assumed to be dominated by the interconnect capacitance, so the constant load capacitance is obtained by careful routing.

SecLib is introduced in 2004 by Sylvain Guilley *et al.* [193]. This logic is based on an quasi-delay insensitive asynchronous primitives, that are balanced to provide constant evaluation and precharge time and dissipation. Specially crafted transistor-level symmetry grants SecLib a higher resistance level to attacks than WDDL, albeit at a high cost in terms of silicon area [188, 189].

In 2005, SABL and "Dynamic Current Mode Logic" (DyCML) [7] are compared by François Macé *et al.* [272]. In DyCML, only one of the output nodes is discharged during the precharge phase. This leads to better performances, such as a reduction by 80 % of the power delay product and by 50 % of the power consumption. In addition, DyCML is assessed to be more resistant to DPA than SABL.

Recently, Francesco Regazzoni *et al.* explore the resistance of "MOS Current Mode Logic" (MCML) against DPA [378] up to simulated attacks. Preliminary results show that MCML has a strong potential for protecting circuits.

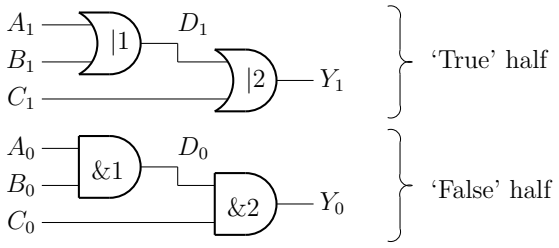


Figure D.3: WDDL testbench in which a data-dependency in the power usage is observed.

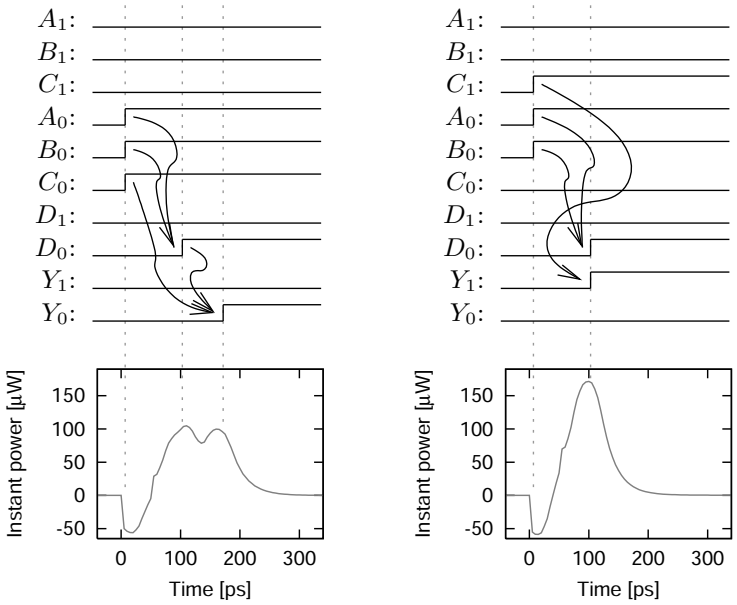


Figure D.4: Power signature that betrays the value of the Boolean variable C, in the setup of Fig. D.3.

D.3.1.2 Early Evaluation Flaw

All the DPL styles presented previously feature a problem mentioned in [439] linked to the intrinsic evaluation in CMOS logic. This logic is memoryless, and thus evaluates as soon as an input changes. Now, in dual-rail logic, the levels of the wires act both as signalization (two wires equal to '0' implies a *precharge* stage), and data (two wires with opposite values mean *evaluation*).

For the sake of illustration, we continue the flaw analysis with the example of WDDL with a `00` spacer to precharge the circuit. This choice makes `OR` gates evaluate faster than `AND`, because `OR` gates simply need one input to have a rising transition to change output values, whereas `AND` gates must wait for two rising transitions to update their output. A scenario that illustrates the data-dependency of the computation flow (and of the power dissipation) is given in Fig. D.3. This testbench shows an `OR3` gate, receiving

its three inputs A, B and C from synchronized registers. The circuit is synthesized in two two-input `OR` gates in cascade. As depicted in Fig. D.4, we assume that the attacker is able to place the circuit in the state $A = B = 0$ (*i.e.* $A_0 = B_0 = 1$ and $A_1 = B_1 = 0$) and tries to guess the value of C by power analysis. The circuit is in precharge state for the negative values of the time t, and the evaluation starts synchronously for all signals at $t = 0$. We observe that, depending on the value of C, the structure of the dissipation differs:

- When $C = 0$, the `AND` gate called &1 evaluates to true (independently of input C, b.t.w.) and, about 50 ps after that, the second `AND` gate called &2 evaluates to one, resulting in two distinct power consumption peaks.
- When $C = 1$, the `AND` gate &1 and the `OR` gate denoted |2, evaluate simultaneously (at first order), which results in a single power peak.

The power signature thus depends on the value of the variable C. Notice that the problem happens because two paths with different delays converge on the same gate, namely &2 in the false network half and |2 in the other. Incidentally, following the *early evaluation*, WDDL also suffers from an *early precharge* symptom in the next clock cycle.

However, it must be underlined that for this bias to be exploited, the attacker must have an acquisition apparatus that is able to detect 50 ps timing variations. In addition, if the acquisition is somehow low-passed filtered, then the difference vanishes. In the SPICE simulations shown in Fig. D.4, the energy consumed by the total transitions is 10.8 fJ for the late evaluation case ($C = 0$) and 11.0 fJ for the early evaluation case ($C = 1$). As these values are very close one from each other, the detection of the difference seams chancy. Nonetheless, the skews add up when descending into the combinatorial logic netlist. A successful attack on a masked DLP (MDPL) circuit exploits a skew of 1 nanosecond at the end of a combinatorial path [358].

In this article, we study SecLib (see Sec. D.3.1.4), a DPL style that does not evaluate early. We compare it with WDDL, because, to the authors' knowledge, it is the only DPL style actually implemented in real cryptographic chips (namely ThumbPod [454] and SCARD [404]). In addition, WDDL does not draw a large current peak at precharge, which simplifies the power planning.

D.3.1.3 Wave Dynamic Differential Logic (WDDL)

WDDL is a DPL implementable with standard cells. Its principle is that, when a Boolean function $f(x)$ is to be computed, its dual $g \doteq \overline{f(\overline{x})}$ is computed in parallel, so as to mask its activity. Provided that the gate is precharged to zero before every evaluation, either f or g has a transition (exclusively), which ensures a power-constant computation. In SecMat v3, the WDDL synthesis was realized based only on `AND` and `OR` instances. Standard cells of several "drive force" from a design kit are armored, so as to:

- ease the pins accessibility and to make pins symmetrical, as required by the backend duplication method, and
- to wrap the standard cells into an electromagnetic cage.

The standard cell is made up of transistors, polarization well-taps and interconnect wires up to the first metal layer (M1). The added coating consists in the superimposition of

stripes of the second metal layer (M2). The steps involved in the construction of the armored `AND` and `OR` gates are detailed in Fig. D.5: the standard cell (1) is added M2 coating (2) to end up with the armored cell (3) = (1) + (2).

D.3.1.4 Secure Library (SecLib)

SecLib is a balanced quasi-delay insensitive (QDI) cells library that enables power-constant and timing-constant computations. The design of each cell involves two stages:

1. a front one in charge of inputs synchronization and
2. a back one in charge of the output computations.

Muller C-elements [414] realize the synchronization task. At this stage, the input is decoded. The second stage consists in the redirection of the value to the adequate output, thanks to `OR` or `XOR` gates. Redundant logic is added to balance the paths to the *direct* (Y_1) and *dual* (Y_0) output couple, resulting in the schematic given in Fig. B.1. In SecLib, The computation is realized for both the direct and its dual output with the same logic, namely a three-input `OR` gate, which provides a protection against an attacker that would be capable of distinguishing the two halves side-channel signature. The use of C-elements increases the cost in terms of area, delay and power consumption of SecLib cells. However, they do fix the "input skew" issue.

Another advantage of SecLib over WDDL is the large range of logic functions that are affordable – security-wise. For instance, as opposed to WDDL, the SecLib gates can be "logically" inverting and non-positive. Indeed, the C-elements of SecLib handle the precharge state; the evaluation is thus unrestricted. The SecLib library includes the following combinatorial cells: $(A, B) \mapsto \{A \cdot B, A \cdot \overline{B}, \overline{A} \cdot B, A \oplus B, A + B, \overline{A} \cdot \overline{B}, \overline{A \oplus B}, A + \overline{B}, \overline{A} + B, \overline{A + B}\}$. This variety of gates helps to reduce the silicon area overhead of SecLib over WDDL [189].

D.3.1.5 Common WDDL and SecLib Cells

The DFFs and the buffers are reused directly from the design kit libraries.

For both WDDL and SecLib, the inverter is implemented as a hard-wired cell, depicted in Fig. D.6. As those two logic styles expect the netlist to be reset to zero during precharge, the inverter cannot be implemented by the application: $(\mathtt{a_{true}}, \mathtt{a_{false}}) \mapsto (\overline{\mathtt{a_{true}}}, \overline{\mathtt{a_{false}}})$. Instead, the wire crossing $(\mathtt{a_{true}}, \mathtt{a_{false}}) \mapsto (\mathtt{a_{false}}, \mathtt{a_{true}})$ is adequate. Consequently, the inverter of Fig. D.6 does not contain any transistor.

The special cells added for WDDL and SecLib synthesis are compatible with standard cells. The height is equal to 12 pitches, divided into a 5-pitch P-well and a 7-pitch N-well.

D.3.2 Placement and Routing

Two standard methods exist to achieve a balanced dual-rail routing. With the **fat wire** [457] technique, the router tool is tricked into seeing one large wire instead of a couple. The conversion from the resulting single-ended to the dual-rail design is done

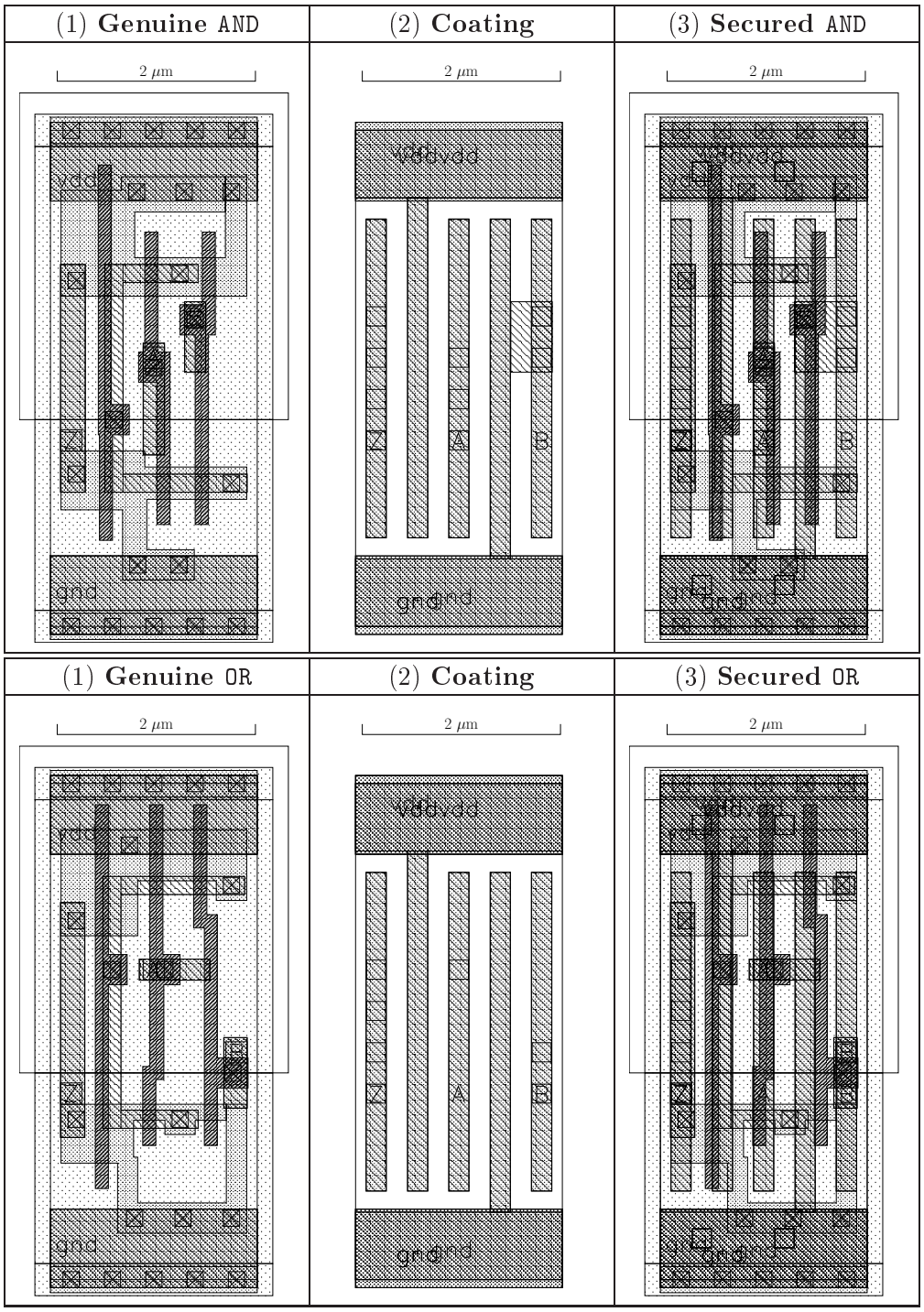


Figure D.5: Two-input logic AND and OR gates armoring, suitable for WDDL.

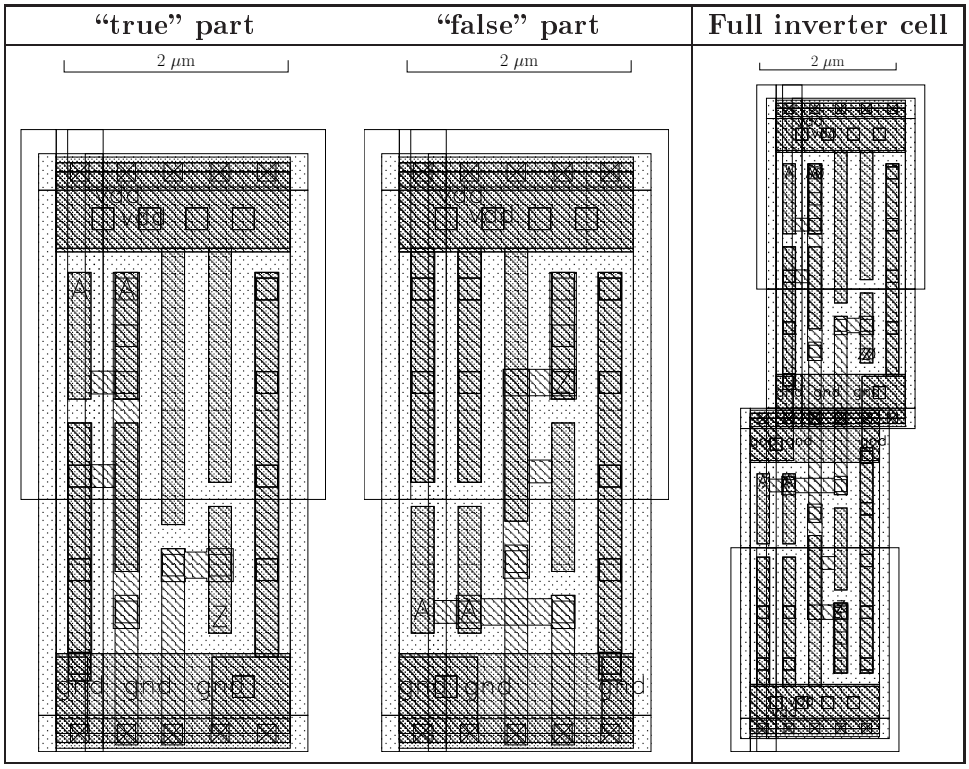


Figure D.6: Logical inverter in dual-rail logic, suitable for both "WDDL" and "SecLib" DPL styles.

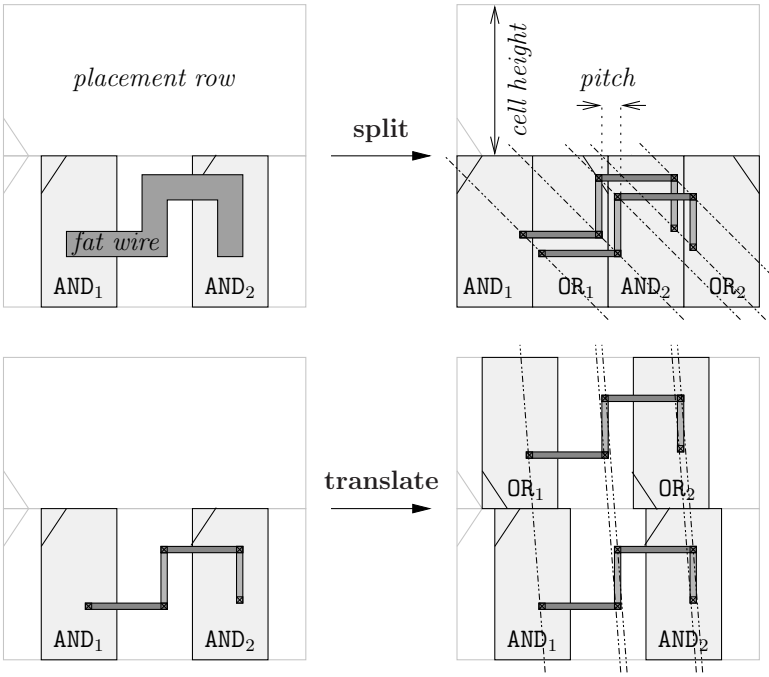


Figure D.7: Fat wire (*upper*) and backend duplication (*lower*) paths balancing illustration.

afterwards by a script. The **"backend duplication"** [191] technique consists in a copy-and-paste of half the design, placed-and-routed (P&R) with half of the resources obstructed, so as to leave room for a subsequent duplication. The true part of the design is first placed every other row. The false part can therefore fit in the free (because firstly obstructed) placement rows. The same strategy is applied to the interconnect: for every level of metallization, half of the routing tracks is blocked. This precaution makes it possible to route the dual nets in the tracks that have been reserved for them, without creating any short circuit with the regular nets. Compared to the fat wire technique, the backend duplication does not require to tamper with design rules used by the P&R tool, because it relies solely on constraints. Although defining routing constraints are sometimes described as "practically too complex", we report here that no more than about two hundred lines of TCL scripts (generated automatically from the floorplan description file) can actually suffice to implement the "backend duplication" technique.

The principles of the two placement and routing methods are illustrated in Fig. D.7. As the access to the pins of the dual-rail gate instances is difficult with the first method, we have opted for the second one.

Both methods can be enhanced by a systematic shielding of the pairs. This option improves drastically the balance of the pairs in each wire couple, albeit at the expense of routability. In our quest to design a DES co-processor as secure as possible, we decided to apply a systematic shield, which resulted in the design being constrained by the wires.

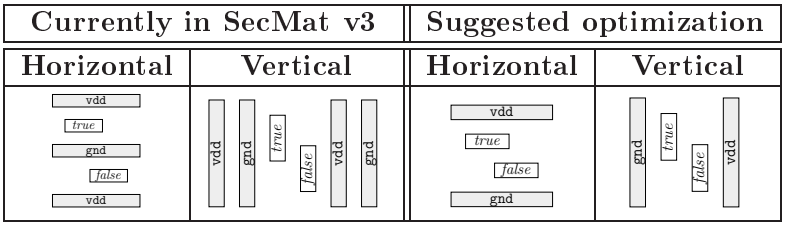


Figure D.8: Horizontal and vertical routing tracks allocation.

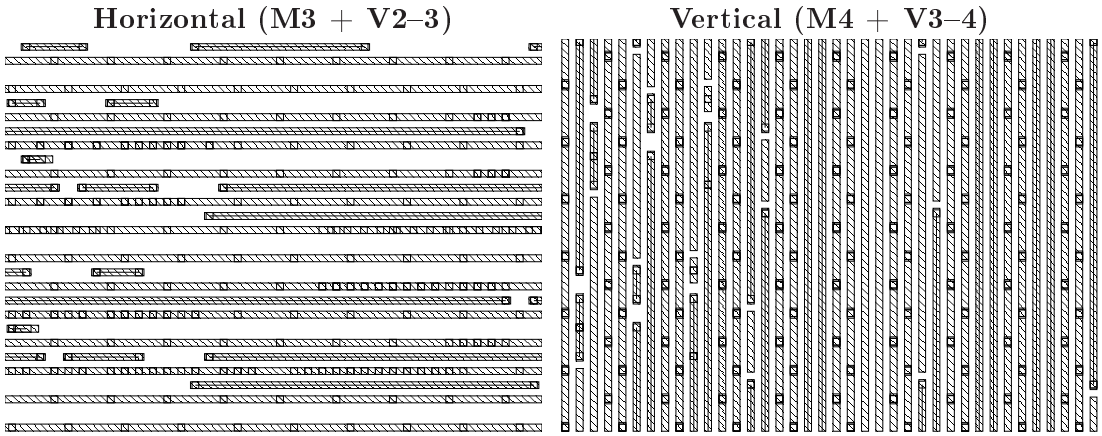


Figure D.9: Typical congested routing zone in the SecMat v3 WDDL DES module.

As discussed in [188], in SecMat v3, the placement density of WDDL (resp. SecLib) is 35 % (resp. 95 %).

The shielding method used for both WDDL and SecLib is based on a periodic routing track allocation depicted in the left part of Fig. D.8. The corresponding layout is illustrated in Fig. D.9 for a typical area of the DES WDDL module. We clearly see that the minimally-sized metal layers are mostly crowded, which is characteristic of a **routing congestion** problem.

The shielding method used in SecMat v3 can be optimized. The number of shielding signals can be divided by two, in both directions, without reducing the insulation between the pairs. The corresponding power planning layout is described in the right part of Fig. D.8. Instead of 4 tracks to route a dual-rail signal, only 3 are now necessary, both vertically and horizontally; this new shielding scheme enables a $100 \times \left(1 - \left(\frac{3}{4}\right)^2\right)$ % silicon area saving. The density of WDDL can thus be increased from 35 % to 62 %. SecLib density is already 95 %: possible silicium savings are not significatively impacted by a new shielding method. Therefore, the overhead for WDDL can be reduced from 11.8 to $6.6 = 11.8 \times (3/4)^2$. This ratio is still twice larger than in the implementation reported in [451].

Table D.1: Performance of SecMat v3 DES modules.

	Reference	WDDL	SecLib
Area [μm^2]	25 368	299 824	382 871
Energy [nJ/encryption]	97.2	2×106	2×197
DES-CBC speed [Mbit/s]	266.7	266.7 / 2	266.7 / 2
3DES-OBC speed [Mbit/s]	88.9	88.9 / 2	88.9 / 2

D.3.3 Performances

Table D.1 reports the performance of the DES modules. The area of the WDDL module is larger than the factor 3 of overhead claimed in [451] because in SecMat v3 every pair of wire is shielded individually. As the dual-rail modules are limited by the routing, it is not surprising that WDDL and SecLib modules have roughly the same area. The power dissipation has been measured experimentally at 8 MHz under the nominal voltage (1.2 volt). It is expressed as the energy per ECB encryption of one 64-bit block. The DES modules were synthesized to run at 66.7 MHz. At this frequency, the regular DES is able to encrypt or decrypt:

- at 266.7 Mbit/s in DES-CBC mode with a 56-bit key, or
- at 88.9 Mbit/s in 3DES-CBC mode with a 112-bit key.

The dual-rail modules operate twice slower, because every computation step is interleaved with a precharge step.

The performance table shows that securing a chip with WDDL or SecLib has definitely a non-negligible impact both on the cost and on the power budget of the cryptoprocessors. However, these co-processors have been designed with the primary goal to resist power attacks. Actually, as proved in the next section D.4, this goal has been reached. Improving the performances while remaining SCA-proof is a challenge we need to address in future research. Second, it must be kept in mind that if the area bloat is undebatably impressive, it can remain acceptable in absolute value. For instance, in the same technology, the 0.3 or 0.4 mm^2 of the secured DES module can be contrasted to an unprotected AES module encrypting an 128-bit block in 44 cycles (0.2 mm^2 [208]) or a 32-kbyte RAM (0.8 mm^2 [208]). Regarding the dissipation, WDDL does not consume much more than twice the power the reference module does (the factor two accounts for the necessary precharge/evaluation dynamic). Roughly speaking, WDDL is built with twice more gates than a single-end logic, but only half of it is activated. SecLib consumes more because each gate is actually made up of several CMOS gates (C-elements followed by `OR`). As compared to WDDL, one can argue this weakens SecLib. However, as explained in Sec. D.4.3, the power consumption is higher but the information leakage it conveys is lower.

D.4 Attacks

We assume that the attacker is able to collect power traces from a circuit. We give the attacker the maximum strength by easing the access to the side-channel and to the synchronization with the encryption. The attacker is fair – it has the same strength irrespectively of the attacked DES module. The exact strength of the attacker is described in the following sections.

D.4.1 Experimental Traces Collection

Given the small spatial extension (a few tenths of square millimeters) of the cryptoprocessors, a local electromagnetic attack (EMA) is not realistic. With standard antennas, the signal collected would be that emitted globally by the DES cryptoprocessor. This brings down the EMA to a powerline analysis. Thus, we decided to focus on power measurements instead.

One typical power trace for each module is shown in Fig. D.10. We measure the differential voltage across a spying resistor, when SecMat v3, running at 33 MHz, performs an ECB encryption of an all-zero message with the key `0x6b65796b65796b65` (*i.e.* "`keykeyke`"). FThe power traces have been averaged 64 times by the oscilloscope, The power traces are averaged 64 times by the oscilloscope, in order to remove the ambient noise and to increase the vertical resolution from 8 to 12 bits.

A typical waveform is shown in Fig. D.11. The trace shows that a static leakage current exists.

The Fourier transform of typical traces for each module is given in Fig. D.12. The clock harmonics (33 MHz) are visible on all spectra. A peak at half the clock frequency is observable for the WDDL version of DES. This frequency is characteristic of the (precharge, evaluation) dynamic, illustrated in Fig. D.11. The reason why the SecLib module does not feature this peak is not intrinsic; it is rather an acquisition artifact, documented in Appendix D.7. In this Appendix, it is shown that this peculiarity does not affect the fairness of the security evaluation of SecLib. In the WDDL spectrum, some additional peaks are visible for multiples of half the clock period (*e.g.* 50, 100 MHz). Beyond 100 MHz, all the three spectra feature the same high-frequency components. Therefore we do not expect to exhibit any special side-channel in the $[100\text{ MHz}, +\infty[$ bandwidth. Consequently, the traces are used plain, without any initial signal processing.

In order to assess the security level of each DES module, we collected 6,400,000 traces for each of them. Gilles Piret suggests in [356] a method to optimize the number of measurements to disclose the key. He basically proposes two complementary ways to accelerate an attack:

1. If the plaintexts are chosen uniformly in front of the attacked substitution box, the selection function bias in the early stages of the correlation attack is minimized.
2. If the plaintexts bits not involved in the sub-key attack are chosen constant, the algorithmic noise is minimized.

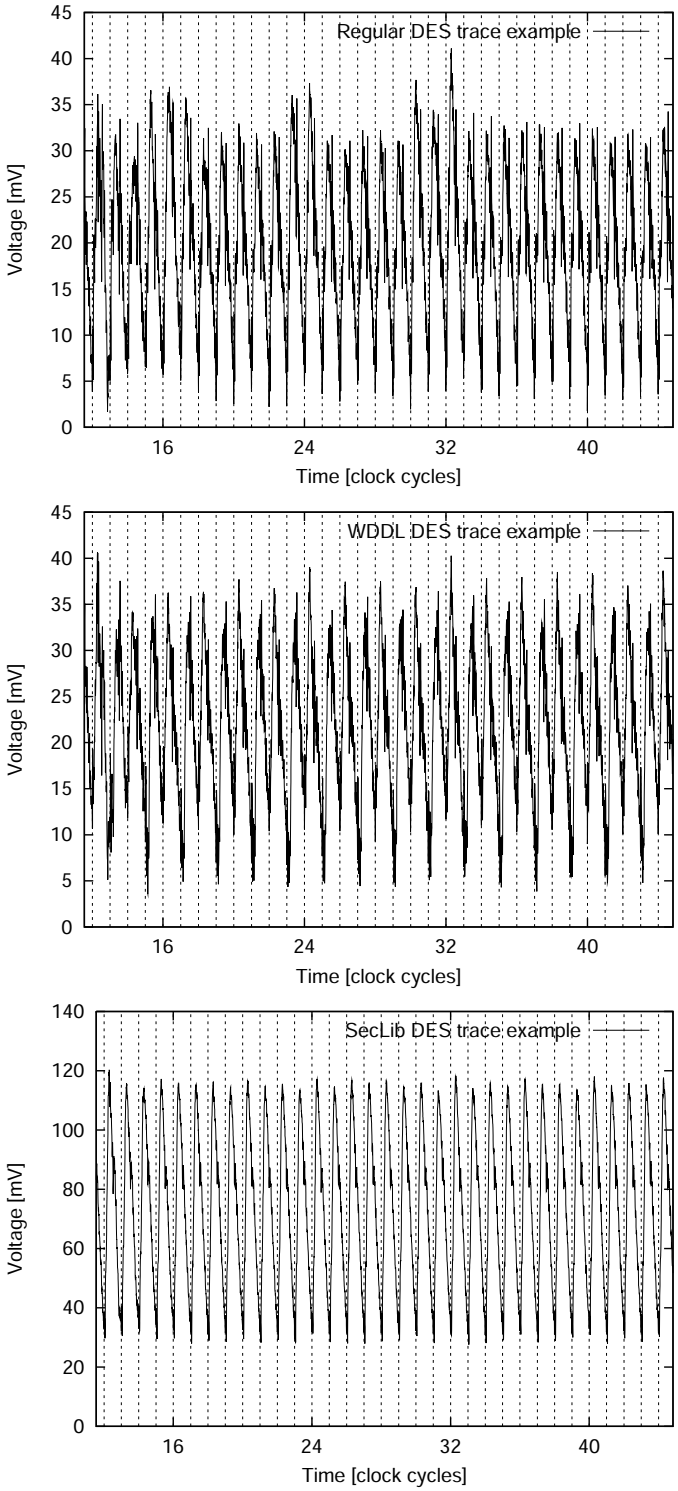


Figure D.10: Regular, WDDL and SecLib DES modules typical instantaneous voltage drops across the 50 Ω spying resistor.

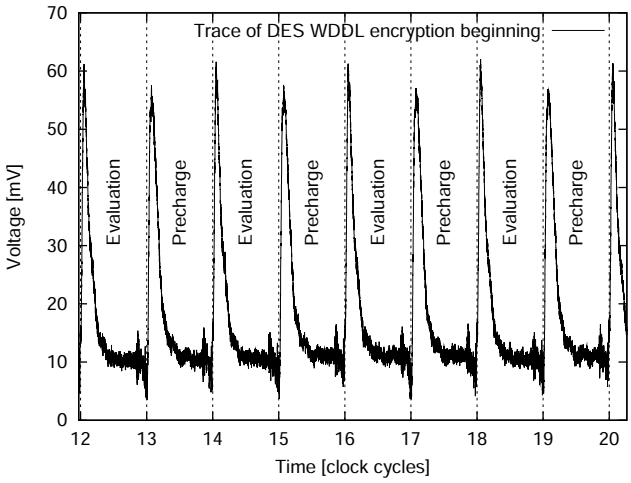


Figure D.11: Under-clocked DES WDDL module current trace.

These two ideas help accelerate an attack, but do not impact its success or the failure with an unlimited amount of side-channel information. As our goal is to test whether the circuits can be asymptotically broken, we simply chose the plaintext randomly with UNIX `rand(3)`.

From a pure cryptographical standpoint, the number of measurements is not large: $6,400,000 \approx 2^{22.6}$, to be contrasted to the $2^{168} = 2^{3\times56}$ number of keys in triple-DES with three independent keys.

However, it can give some insights about how much security is available in hardware: it lets the security strategy be partitioned into a hardware/software mixture. For instance, in the context of stream encryption with DES in CFB, OFB or GCM modes of operation, it can give an indication on the frequency of keys renewal: diversified keys regenerated at the rate of one per $6,400,000$ encrypted blocks is enough.

D.4.2 Off-line Attack on the Reference DES Module

In this subsection, efforts are devoted to identify the strongest attacks against the reference DES module. The incentive is to define the best analyses suitable for the protected instances, discussed in the forthcoming subsection D.4.3.

D.4.2.1 Description of the Power Attacks

It is customary to divide power attack into two classes:

i) mono-variate analyses, such as IPA, DPA or CPA, and

ii) multi-variate analyses, such as template attacks.

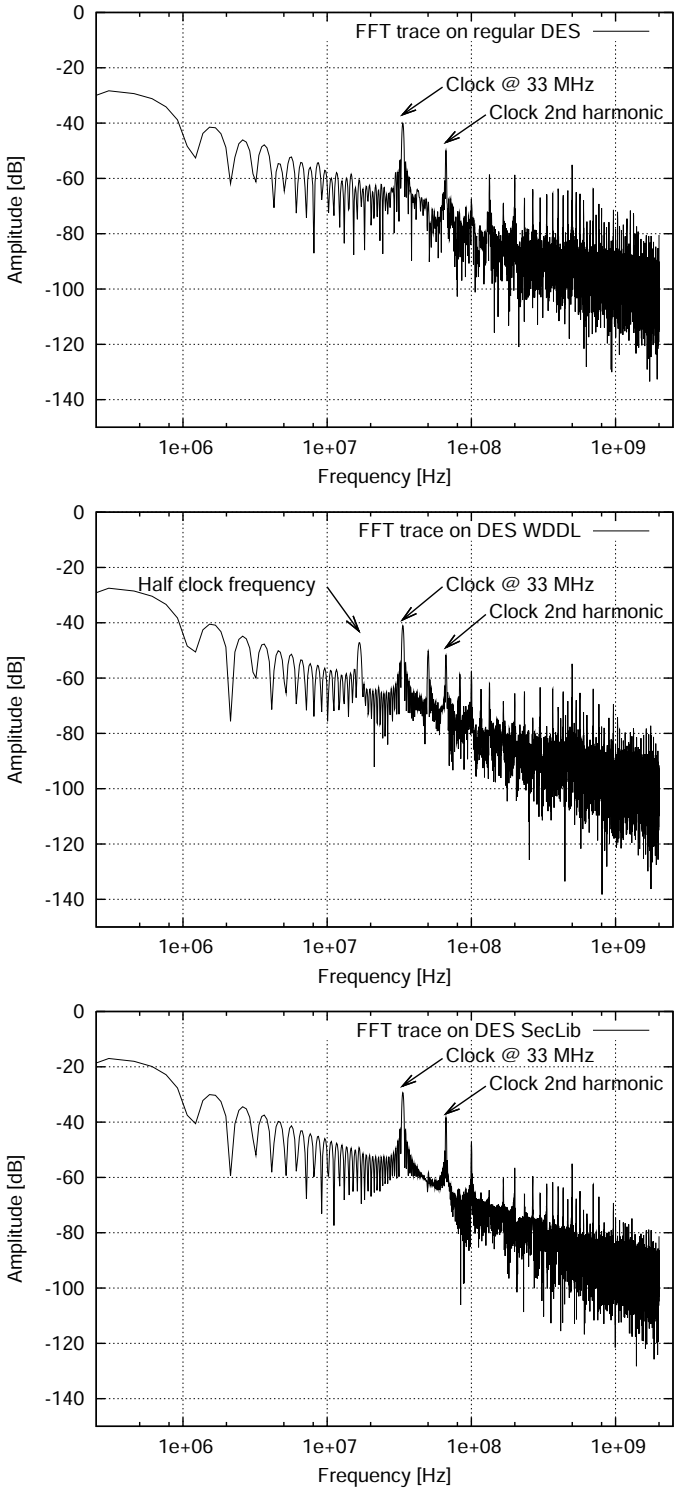


Figure D.12: FFT of three power traces from regular, WDDL and SecLib DES modules.

D.4.2.1.1 Correlation Attacks. We discard IPA because it is too unfavorable from the attacker viewpoint and too specific (it targets software implementations). Instead, we wish to describe the most powerful attacker against a hardware parallel implementation.

Other mono-variate attacks can be nicely unified by the enhanced CPA [258], a heuristic technique that bridges the gap between CPA and DPA. For each side-channel instant, it consists in computing a biased correlation coefficient between the acquired trace (denoted W, as in waveform) and the expected dissipation (denoted H, as in Hamming weight or distance).

If W and H are considered random variables, we note $\mathsf{E}W$ the expectation of W and $\sigma_W \doteq \sqrt{\mathsf{E}\left(W - \mathsf{E}W\right)^2}$ its standard deviation (idem for H). The covariance between W and H is defined by: $\mathsf{Cov}[W,H] \doteq \mathsf{E}\left((W - \mathsf{E}W)\cdot(H - \mathsf{E}H)\right) = \mathsf{E}\left(W \cdot H\right) - \mathsf{E}W \cdot \mathsf{E}H$. The correlation factor between W and H is the normalized quantity, constructed as: $\rho_{W,H} \doteq \frac{\mathsf{Cov}[W,H]}{\sigma_W \cdot \sigma_H}$. The Cauchy-Schwarz theorem implies that the correlation factor is normalized:

$$-100\ \%\ \leq\ \rho_{W,H}\ \leq\ +100\ \%\,.$$

The H random variable is actually parametrized by a sub-key to guess. In DES, the dissipation can be split into eight contributions, each of which corresponding to the substitution boxes (sbox) layer. In each sbox, 6 bits of the key are mixed with the datapath, both at the first and at the last rounds. We thus end up, for every sbox (there are 8 of them in DES), with 2^6 H functions.

The DPA consists in guessing the key according to the greatest value of $\mathsf{Cov}[W,H]$, when H explore all the possible key guesses weighting functions. The resulting waveforms are called differential traces, and consist in the extraction of a selected dissipating phenomenon from the overall crypto-processor power consumption.

The CPA [60] simply differs from the DPA in that it uses the correlation factor $\rho_{W,H}$ instead of the plain correlation $\mathsf{Cov}[W,H]$ to choose which key candidate is the best. It is customary to designate CPA by the term DPA, and to distinguish them as "correlation-based" or "distance of mean" for the classical one.

The enhanced CPA introduces an empirical parameter $\varepsilon \in [0, +\infty[$. The correct key decision is made based on the biased parameter comparison for the 64 key guesses:

$$\frac{\mathsf{Cov}[W,H]}{(\sigma_W + \varepsilon)\cdot\sigma_H}\,.$$

For $\varepsilon = 0$, the enhanced CPA is equal to the regular CPA. When $\varepsilon \to +\infty$, and provided σ_H is not noisy (for instance using the chosen plaintext methodology described in [356]), the contribution of σ_W is cancelled and the enhanced CPA tends towards the DPA.

The empirical ε offset makes up for a possible statistical artifact: the uninteresting instants in the power curves also correspond to the minimal variance σ_W. However, if this value is too low, $\rho_{W,H} \propto 1/\sigma_W$ becomes artificially large; there is thus the risk that an automatic peak detection software be fooled by such a spurious peak. As on our measurements $\sigma_W > 2.5$ mV, the protection offered by ε is useless.

D.4.2.1.2 Template Attacks. Template attacks [69] consist of a two-phase strategy. First, a probabilistic model of the dissipation is built based on the training on a clone device. Second, an intercepted trace is matched against the pre-characterized templates. The practical problem raised by template attacks is the high dimensionality of the data used in the training phase. To alleviate the memory and computational requirements, Archambeau *et al.* [13] proposed to use the principal components analysis (PCA [231]). In many concrete cases, PCA is appropriate. The basic assumption made in PCA is that all templates share a common diagonalization basis; it has been shown to be realistic in many cases.

Unlike correlation attacks (DPA or CPA), that target a single sample in the traces, templates with PCA collect a distributed leakage. Indeed, PCA constructs a linear combination of samples that maximizes the variance (dependency in the key). This analysis is thus able to capture the skews induced by the early evaluation problem of un-synchronized DPL styles, such as WDDL.

D.4.2.1.3 Vulnerability Metrics. The two attack classes just presented allow to qualitatively compare two implementations. If one implementation is broken by an analysis and not the other, then the former is weaker than the later.

However, in the case where two implementations resist an attack[1], correlation and template analyses can produce quantitative metrics that reflect the intrinsic degree of vulnerability of an implementation. For such a vulnerability estimator to enable security comparisons, it must be homogeneous for the various implementations to compare.

We propose three homogeneous metrics that are proportional to the vulnerability criticality.

The first metric is the amplitude of the DPA peak. In [196], it is shown that the differential traces are the extraction of a relevant part from the chip's overall activity. The targeted logic gates are identified by the DPA selection function. This quantity is thus expressed in the units of the side-channel measurement. As we use a differential voltage probe, the side-channel unit is the volt. This metric might not be appropriate for two unrelated experiments, with different acquisitions apparatuses and conditions. However, the SecMat v3 architecture has been devised to enable comparisons: the side-channel is measured from the same power pads, with the same probe and the same oscilloscope setup.

The second metric is the best correlation factor obtained by CPA. This metric does not have any unit, because it is a ratio. The correlation factor also allows to compare two different setups, since it is relative to the acquisition noise (σ_W).

Finally, the third metric is the largest eigenvalues obtained by template attacks in PCA. Its interpretation is the maximal variance (dependency in the secret) that can be extracted from the side-channel. The units of the eigenvalues are the square of the side-channel, because they represent the square of a standard variation. Thus, as already discussed for the first metric, they are applicable only to setups designed specifically to

1. This happens to be the case for WDDL & SecLib modules (see Sec. D.4.3).

Table D.2: Number of traces required to attack the reference DES co-processor with DPA and CPA.

Sbox Index	First round		Last round	
	DPA	CPA	DPA	CPA
#1	146,368	163,008	92,480	65,024
#2	183,040	206,080	201,920	146,816
#3	263,296	227,456	109,440	96,640
#4	191,360	149,376	84,608	72,192
#5	160,384	136,256	79,680	81,984
#6	92,992	89,856	32,000	18,304
#7	241,152	247,552	47,744	47,808
#8	41,280	37,888	227,840	191,744
Worst	263,296	227,456	227,840	191,744
Best	41,280	37,888	32,000	18,304

enable comparisons. It is thus relevant for the comparison of the three SecMat v3 DES modules.

D.4.2.2 Attack Results of the Reference DES Module

The reference DES module is easily broken with both DPA and CPA. The number of measurements to disclose (MTD) the key is given in Tab. D.2. The CPA appears to be the best attack on average. We provide in Fig. D.13 the correlation factors obtained after 80k traces accumulations.

We tried the enhanced CPA. This technique is supposed to improve the speed of the CPA; however, apart from sbox #2, the gain is marginal or null, and sbox-dependent. As the protected DES modules have different sboxes (synthesis and P&R differ), the improvement is not expected to be portable. The results are given in Fig. D.14.

The thorough analyses made on the reference DES module led to conclusions stated in Tab D.3. Based on these results, we can motivate a trustworthy model of an empowered attacker against the two protected instances. To summarize the information gained by an adversary from the preliminary tests, we can say that:

- current traces are preferred over electromagnetic traces,
- traces are used without preprocessing,
- regular CPA or templates with PCA are definitely the best attacks,
- the correlation attacks are slightly better on the last round than on the first one. However, for reasons disclosed in Appendix D.6, the attack on the last round is more subtle. Therefore, in order to present unambiguous results, the attacks are performed on the first round.

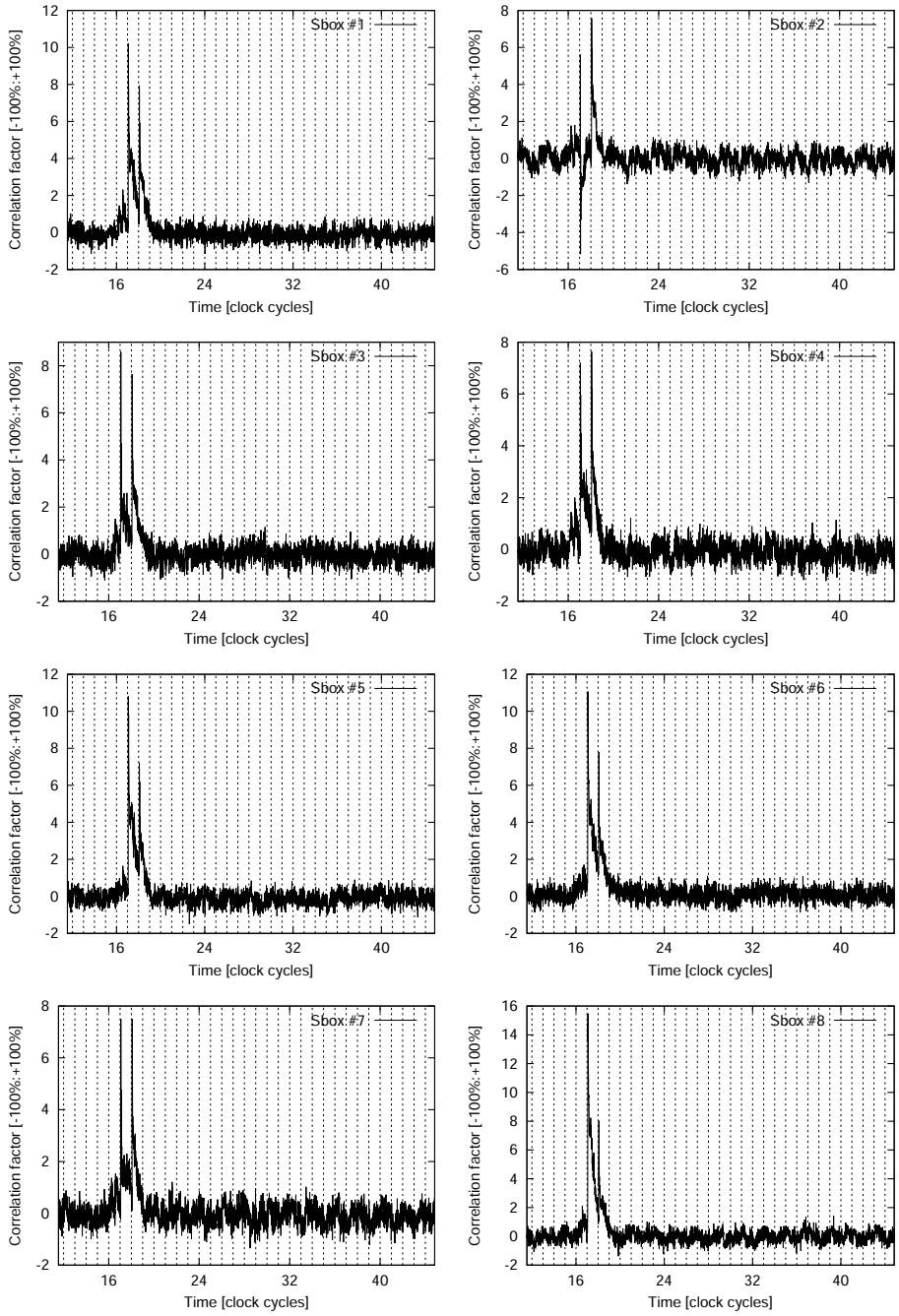


Figure D.13: Correlation factor for the correct key guesses obtained when attacking the first round of the reference DES module's eight sboxes.

Table D.3: Analysis of the attack strategies relevant for SecMat v3.

Attack	Relevance	Description
SPA	no	The control of DES is data-independent
IPA	no	Less powerful than CPA
DPA	no	Less powerful than CPA
CPA	yes	Appropriate
Enhanced CPA	yes	But the improvements are not statistically representative
Templates with PCA	yes	Eigenvalues describe the optimal dependency on the key

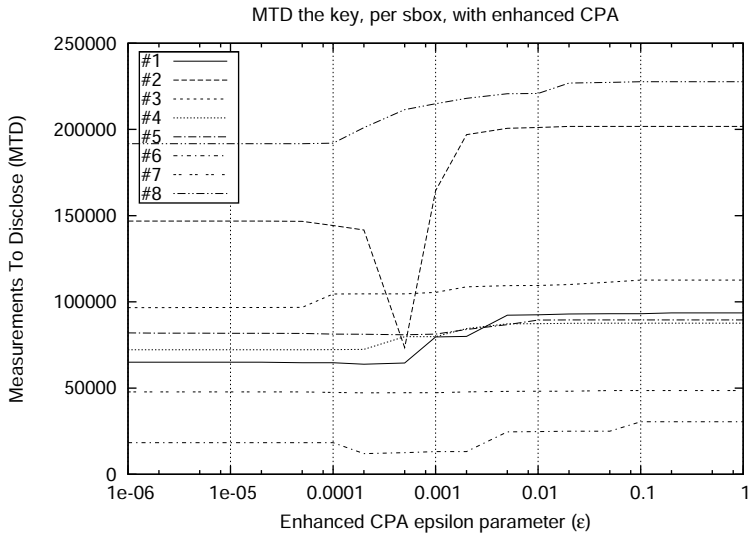


Figure D.14: MTD on the reference DES module, attacked with enhanced CPA on the last round (for the eight sboxes).

Table D.4: Extremal correlation factors of CPA on the first round of WDDL and SecLib DES.

Sbox index	DES WDDL		DES SecLib	
	Min.	Max.	Min.	Max.
#1	-1.10 %	+1.10 %	-5.3 %	+4.2 %
#2	-0.82 %	+0.84 %	-5.2 %	+6.6 %
#3	-0.87 %	+1.00 %	-5.2 %	+6.5 %
#4	-0.90 %	+1.10 %	-5.0 %	+6.7 %
#5	-0.93 %	+1.20 %	-6.5 %	+3.9 %
#6	-1.00 %	+1.00 %	-4.7 %	+5.4 %
#7	-1.00 %	+0.95 %	-5.3 %	+5.3 %
#8	-1.20 %	+1.30 %	-7.2 %	+7.8 %

D.4.3 Off-line Attack on the Protected DES Modules

The CPA has been realized on the first round of the WDDL and SecLib DES modules. The only difference between this CPA and the one used for the regular DES is the switch from the Hamming distance to the Hamming weight selection function. Indeed, because of the precharge, the reference state is plain zero. The Hamming distance, as a tool to count transitions, thus degenerates into a Hamming weight.

The correct key fails to be found by the CPA with 6,400,000 traces. The extremal (minimal and maximal) correlation factors over the whole trace (5,000 points) found for the two protected instances are reported in Tab. D.4. It must be emphasized that none of these extremal values correspond to the correct key guess. To illustrate this fact, we show in Fig. D.15 how the correlation power analysis on WDDL and SecLib is erring.

The whole correlation traces are shown in Fig. D.16 for the first sbox. The highlighted trace corresponds to the correct key guess; the others, superimposed in the background, are those obtained by an erroneous key hypothesis. The correlation traces for the other sboxes are similar: no significant peak appears at the encryption beginning (clock period 16).

The template construction results are shown in Tab. D.5. Principal component analysis [13] is used to quantify the amplitude of the variances. The WDDL implementation has two significant eigenvalues, whereas SecLib does not have any overwhelming eigenvalue. The dispersion of WDDL, compared with that of SecLib, after 6,400,000 traces is about $15 = \sqrt{181.2\ \mathrm{mV}^2/0.8\ \mathrm{mV}^2}$. This figure means that the WDDL traces depend on the key about one order of magnitude more than the SecLib traces.

Despite the high values taken by WDDL eigenvalues, the matching of an unseen trace does not work. This can be understood by the fact that the templates quality is not sufficient after their estimation with 6,400,000 traces. To give an idea on the speed

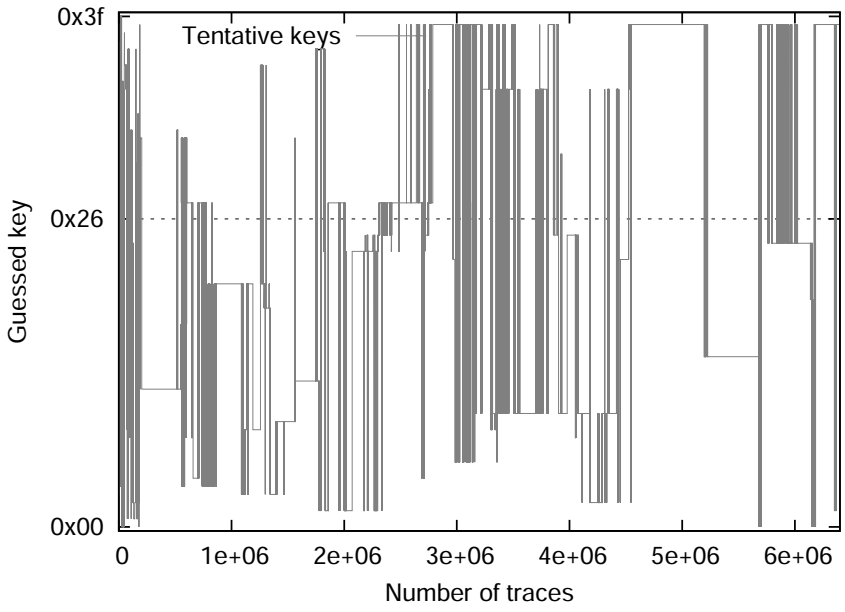


Figure D.15: Key automatically selected by CPA on SecLib DES (correct key: `0x26` $\in$ [`0x00`, `0x3f`]).

Table D.5: Three principal eigenvalues, expressed in μV^2, for the template on the sboxes inputs.

Sbox index	**WDDL**			**SecLib**		
	λ_0	λ_1	λ_2	λ_0	λ_1	λ_2
#1	178.3	22.5	0.3	1.0	0.5	0.3
#2	171.5	20.8	0.3	0.9	0.5	0.3
#3	153.6	17.5	0.2	0.8	0.4	0.2
#4	201.5	21.0	0.4	0.8	0.4	0.2
#5	196.7	17.0	0.3	0.7	0.3	0.2
#6	194.8	14.3	0.3	0.7	0.4	0.2
#7	171.4	18.9	0.3	0.8	0.5	0.2
#8	182.3	20.0	0.3	0.9	0.5	0.2
Average	181.2	19.0	0.3	0.8	0.4	0.2

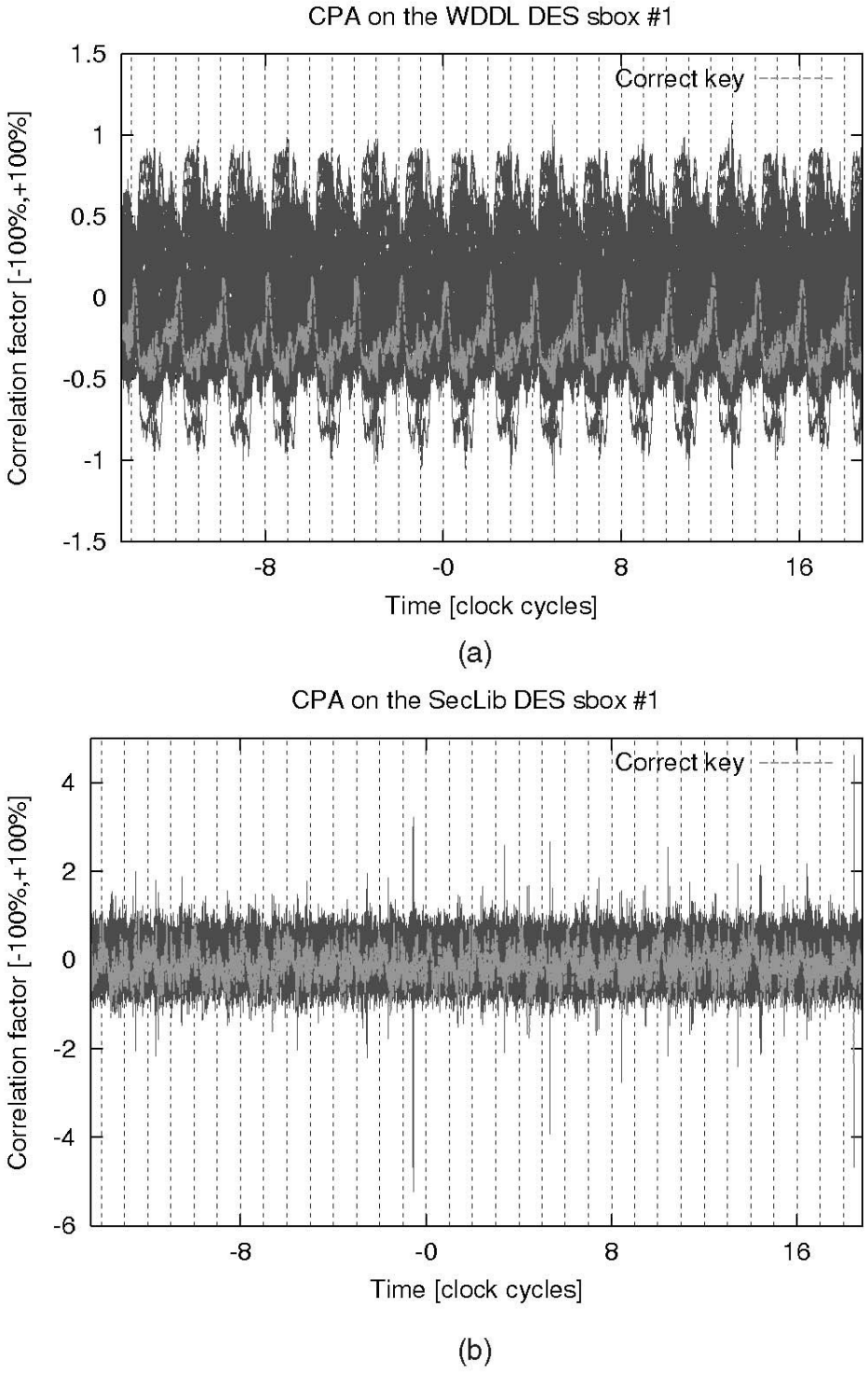


Figure D.16: Trace of the correlation factor for WDDL (*top*) and SecLib (*bottom*).

of the dispersion convergence, the evolution of the largest eigenvalue with the number of traces used to build the templates is given in Fig. D.17. Indeed, the templates are in practice empirical estimators, whose variance decreases with the number of samples used to build them.

D.4.4 Comparison with the State-of-the-Art

The results obtained on SecMat v3 are compared with the state-of-the-art attacks on circuits protected against SCAs in Tab. D.6. The resistance is evaluated with the number of measurements to disclose (MTD) one bit of the key. It can be misleading to compare the MTD the key between different circuits, because setups, acquisition conditions, target algorithms, attacks, *etc* may all differ. We quantify the **security gain** as the ratio between the MTD of a protected and unprotected modules. The selected results have all been validated in silicon. They are listed chronologically.

In 2004, the ADIDES family of asynchronous QDI circuits in 0.18 μm technology (ASIC1) has been successfully attacked [55] because the backend was unbalanced. In 2005, the ThumbPod synchronous power-constant WDDL circuit with parallel routing (ASIC2), implemented in 0.18 μm technology, leaks some key bytes [454]. Possible reasons could be the early evaluation problem or an insufficient wires shield against cross-talk. In 2005, a SoC, realized in 0.25 μm technology, embedding various AES processors protected with algorithmic masking (ASIC3) is broken by correlation analysis [294]. The selection function targets glitches in the sboxes [293]. The two masking schemes are that of M.-L. Akkar [6] (ASIC3.1) and of E. Oswald [346] (ASIC3.2). In 2007, the 0.13 μm SCARD [404] evaluation circuit (ASIC4), containing, amongst others, one reference 8051 CPU and seven protected versions, plus some AES hardwired co-processors, is evaluated. The MDPL [359] version of the 8051 is broken because of the early evaluation issue [358]: the `MOV` instruction leaks the transferred data. Also in 2007, an attack on the SCARD circuit suggests that the MDPL version of AES has a serious breach, due to flaws in the assumptions made on the the randomness source [139]. However, the practicability of this attack is still uncertain, notably because no indication about the number of power measurements to break the implementation is mentioned. Finally, the WDDL (ASIC5.1) and SecLib (ASIC5.2) DES co-processors of the SecMat v3 system-on-chip, the 0.13 μm circuit described in this article, remain unbroken.

D.5 Conclusion

A prototype ASIC, called SecMat v3, has been designed and fabricated in 0.13 μm technology. Its purpose is to evaluate the security level of DES co-processors implemented in two power-constant logic styles: WDDL and SecLib. WDDL is subject to a security flaw: under some circumstances, for instance when a skew exists between two signals, the computation duration does depend on some intermediate data. The SecLib logic features a synchronization stage that prevents early evaluation: in addition to being power-constant, SecLib is also timing-constant. The maximal level of efforts has been

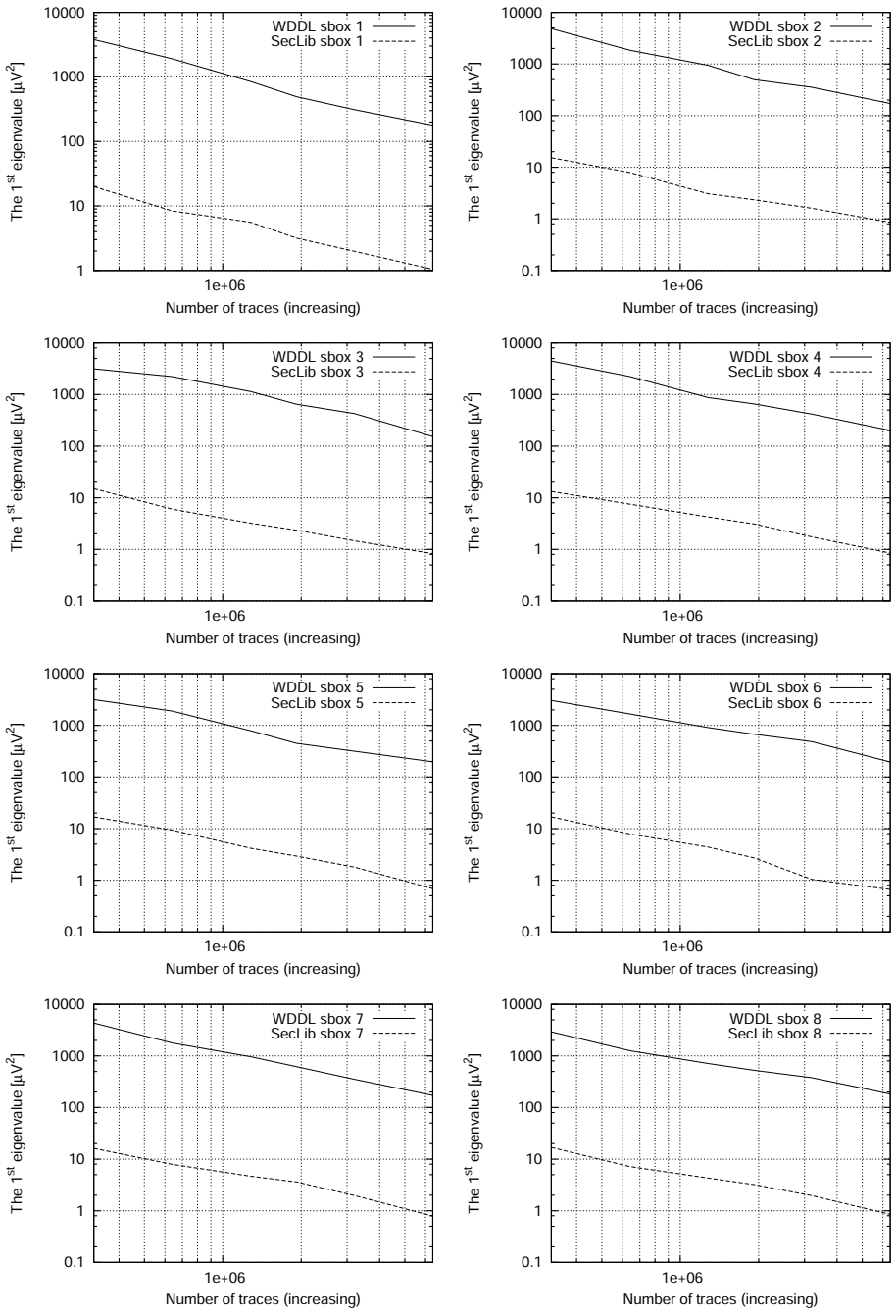


Figure D.17: Decay of the largest eigenvalue for WDDL and SecLib modules when characterized by PCA.

Table D.6: Resistance assessment of protected ASICs, based on real attacks.

Circuit id.	Algo--rithm	MTD		Security gain
		Unprotected	Protected	
ASIC1	DES	10,000	200,000	20.0
ASIC2	AES	320	21,185	66.2
ASIC3.1	AES	25,000	30,000	1.20
ASIC3.2	AES	25,000	130,000	5.20
ASIC4	CPU	279	471	1.69
ASIC5.1	DES	18,304	6,400,000 is not enough	> 350
ASIC5.2	DES	18,304	6,400,000 is not enough	> 350

spent to obtain an accurate idea of the resistance of the protected DES instances. The circuit's architecture, thanks to a power management IP that controls modules clock-gating, allows for fair comparisons of side-channel measurements. The protected modules are carefully designed, especially at the backend level: dual-placement, parallel routing and systematic wire shielding techniques have been used for both WDDL and SecLib modules.

We have found that both secured DES modules feature biases, but that they fail to be exploited by an attack. This does not mean that the DES protected modules are invulnerable. It merely implies that some yet-to-discover attack might defeat them, but that with nowadays attacks, they resisted all our assaults. As of today, the "SecMat v3" ASIC is the most robust power-constant cryptographic implementation because its security gain is the largest published so far (> 350).

Acknowledgements

We are grateful to the anonymous reviewers for their help in improving the presentation of the results and in suggesting a way to reduce the area overhead of the WDDL version of DES. We also wish to thank Florent Flament for designing the SecMat v3 ASIC, Karim Benkalaia for producing the PCB, Jean-Luc Danger and Yves Mathieu for assistance with CAD tools and Ronan Keryell for valuable comments on the project in general. Sumanta Chaudhuri has brought an inestimable help in the specification and the validation of the SecMat v3 ASIC; his encouragements were very beneficial to the success of this "trusted computing" project. This work has been partly financed by the french conseil régional "Provence Alpes Côte d'Azur" (Région PACA).

D.6 Appendix 1: CPA on the last round of the DES modules

The architecture of the three DES modules of SecMat v3 has a peculiarity, that makes the correlation analyses on the last round very singular. We recall that DES is a Feistel cipher, that iterates sixteen rounds. The datapath is divided into two halves, referred to as L and R (standing for Left and Right). For all round, indexed by an integer $i \in [1, 16]$, the datapath computes:

$$\begin{cases} \mathrm{L}_i &= \mathrm{R}_{i-1}\,, \\ \mathrm{R}_i &= \mathrm{L}_{i-1} \oplus f(\mathrm{R}_{i-1}, \mathrm{K}_i)\,. \end{cases}$$

As it can be seen in Fig. D.18, the datapath register LR has no enable, whereas the keypath register CD has one [195, Fig. 6]. Therefore, as DES modules are designed to process blocks of data without dead cycles, at the end of the first encryption, the datapath starts a new one. But, since the key scheduler is disabled, this encryption is done with a constant key for all next rounds. This constant key corresponds to the key of the first round of the first encryption. As a consequence, the contents of LR evolves as shown in Tab. D.7. We recall that, by convention, the encryption starts at clock cycle 16 and ends at clock cycle 32.

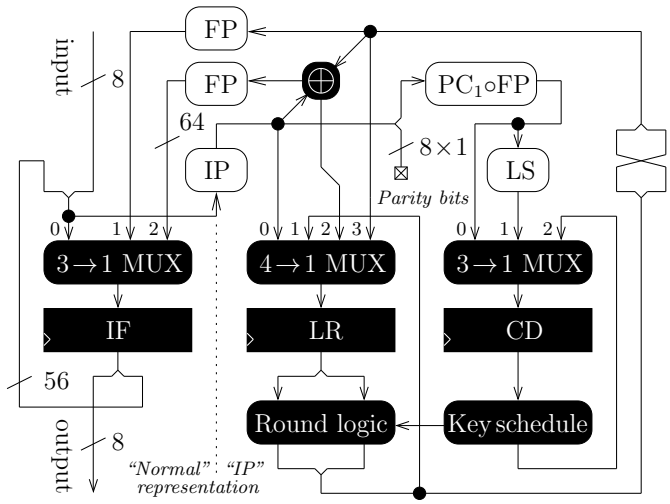


Figure D.18: SecMat v3's multi-modes pipelined DES datapath.

The CPA on the last round uses, for each sbox, the following selection function: $\mathrm{L}_{15} \oplus \mathrm{L}_{16}$. When the last round key K_{16} is unknown, the 64 selection functions, parametrized by K, are computed:

$$\mathrm{L}_{16} \oplus \mathrm{R}_{16} \oplus f(\mathrm{L}_{16} \oplus \mathrm{K})\,, \tag{D.1}$$

where L_{16} and R_{16} are the known ciphertext halves and f is the Feistel function of DES.

Table D.7: Datapath contents in all DES modules of SecMat v3 around the encryption end.

Clk #	Register L	Register R	Comment
⋮	⋮	⋮	
30	L_{14}	R_{14}	Regular round (#14)
31	L_{15}	R_{15}	Regular round (#15)
32	R_{16}	L_{16}	No swap in last round
33	L_{16}	$R_{16} \oplus f(L_{16} \oplus K_1)$	"Encryption goes on"
34	$R_{16} \oplus f(L_{16} \oplus K_1)$	don't care	"Encryption goes on"
⋮	⋮	⋮	

This quantity is correlated to the reference DES power traces. The resulting 64 correlation factor waves are shown in Fig. D.19.

One can easily see that not only the correct key guess causes a correlation peak, but also a key that happens to be the first round key (false correlation peak). This behavior is not observed in the second sbox, merely because it happens that the 6-bit subkey of K_1 is, by chance, equal to that of K_{16}.

The explanation is as follows:

1. **At clock period 31**, there is the transition $R_{14} \rightarrow R_{15}$ in register R. Therefore, the trace is correlated with $R_{14} \oplus R_{15} = L_{15} \oplus L_{16}$. This correlation matches (D.1) when the key guess is correct, *i.e.* when $K = K_{16}$.
2. **At clock period 33**, the transition $L_{16} \rightarrow R_{16} \oplus f(L_{16} \oplus K_1)$ happens in R. The dissipation is correlated with $L_{16} \oplus R_{16} \oplus f(L_{16} \oplus K_1)$, that matches (D.1) when $K = K_1$.
3. **At clock period 34**, the same transition takes place in register L, hence an echo of the previous strong correlation with $K = K_1$.

Consequently, the DES module embedded in SecMat v3 leaks two non-overlapping 6-bit sets of the key when analyzed by a correlation attack on the last round. From an attacker viewpoint, it is thus profitable to restrict side-channel acquisitions to the clock periods [31-34], because the signal is more intense here than during the encryption beginning.

D.7 Appendix 2: Details about Synchronization

In our setup, the encryption is announced by a trigger signal. The CPU of SecMat v3 executes the snippet of code given in Fig. D.20.

Due to the system-level VCI [8] management, the encryption is starting few cycles (deterministic value) after the rising edge of a `PO` signal.

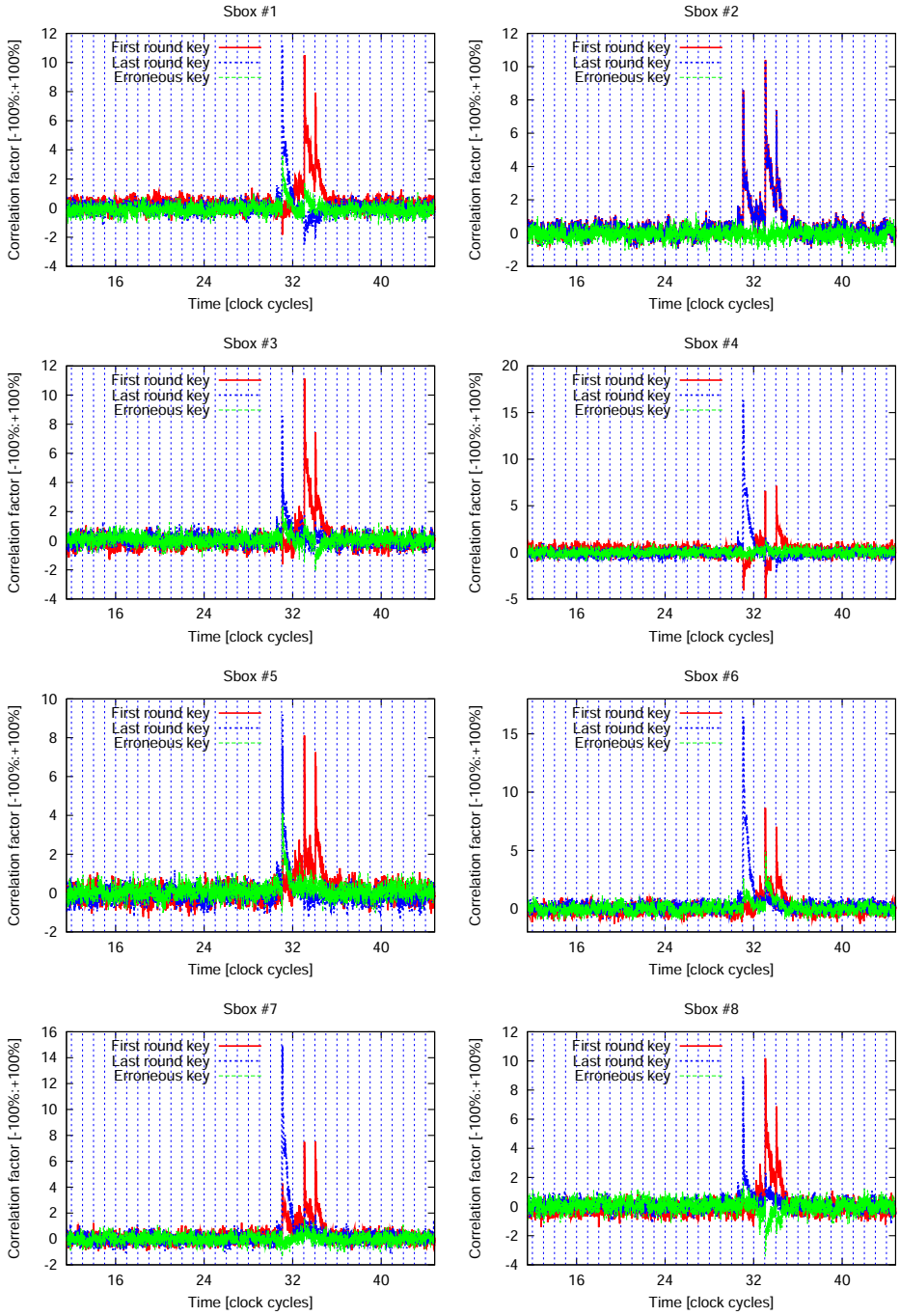


Figure D.19: Correlation factors obtained when attacking the last round of the reference DES module's eight sboxes.

```
for/*ever*/(;;) // Go!
{
   // This block must be executable at least once,
   // otherwise the trigger is skipped and the
   // message is never encrypted:
   do
   {
      memcpy( msg_addr, msg_backup, msgSize );
      // The synchronization signal for the 54622D
      // oscilloscope is PO[0]. The rising edge of
      // PO[0] announces the next encryption:
      PO_write( 0x01 );
      launch_cipher();
      PO_write( 0x00 );
   }
   while( !UART_is_char_in() );
   // Ciphered message is in memory at the plain
   // message's address when exiting.
   switch( UART_get_char() )
   {
      case EXIT: return 0;
   }
}
```

Figure D.20: Code in C programming language executed by the on-chip CPU (VCI master) to realize side-channel acquisitions.

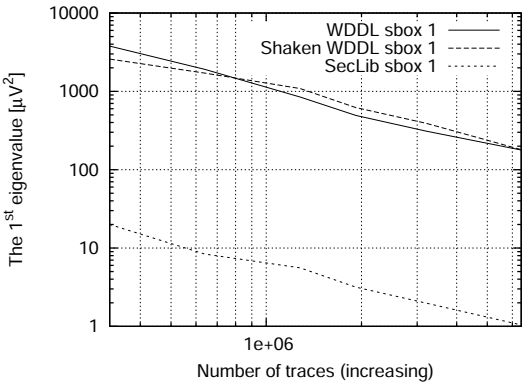


Figure D.21: Decay of the largest eigenvalue for "consistent" and "artificially shaken" WDDL, compared to SecLib, while characterizing protected DES modules by templates in PCA.

However, the dual-rail modules have their own dynamic. During the precharge stage, they cannot accept data to start a new computation. If they receive all the same a request, it is delayed by one clock period for it to arrive in the evaluation stage.

The behavior of the circuit executing the abovementioned code has been simulated under Mentor Graphics ModelSim. It happens that:

– The WDDL module always starts after 26 cycles,

– The SecLib module starts one encryption over two after 25 cycles or after 26 cycles.

Thus, when averaging the signals 64 times, the SecLib DES is accumulating 32 traces starting on time with 32 traces starting one clock earlier. This explains why the FFT spectrum of SecLib (Fig. D.12) does not show a peak at half the clock frequency. We have captured an unaveraged trace of SecLib, and for this signal, the FFT does show the peak that vanished because of the averaging. This "on-chip communication" problem can be safely ignored for our analyses. To bring an experimental evidence to this assertion, we simulated the timing offset on WDDL. The WDDL traces were split into two groups, the second being additionally offset by one clock period. Of course, this helps neither CPA nor DPA. In the template attacks, that are multi-variate, the temporal position of the leak does not affect the results. Indeed, the results shown in Fig. D.21 for sbox #1 confirm this assumption; the other sboxes exhibit the same independence w.r.t. the probabilistic timing offset.

Appendix E

Evaluation of Power-Constant Dual-Rail Logics Counter-Measures against DPA with Design-Time Security Metrics

Extended version of article [210]

Authors: *Sylvain Guilley, Laurent Sauvage, Florent Flament, Vinh-Nga Vong, Philippe Hoogvorst and and Renaud Pacalet*

Abstract

Cryptographic circuits are nowadays subject to attacks that no longer focus on the algorithm but rather on its physical implementation. Attacks exploiting information leaked by the hardware implementation are called side-channel attacks (SCA). Amongst those attacks, the differential power analysis (DPA) established by Paul Kocher *et al.* in 1998 represents a serious threat for CMOS VLSI implementations. Different countermeasures that aim at reducing the information leaked by the power consumption have been published. Some of these countermeasures use sophisticated backend-level constraints to increase their strength.

As suggested by some preliminary works (*e.g.* by Huiyun Li from Cambridge University), the prediction of the actual security level of such countermeasures remains an open research area. This article tackles this issue on the example of the AES SubBytes primitive. Thirteen implementations of SubBytes, in unprotected, WDDL and SecLib logic styles with various backend-level arrangements are studied. Based on simulation and experimental results, we observe that static evaluations on extracted netlists are not relevant to classify variants of a countermeasure. Instead, we conclude that the fine-grain timing behavior is the main reason for security weaknesses. In this respect, we prove that SecLib, immune to early-evaluation problems, is much more resistant against DPA than WDDL.

E.1 Introduction

Side-channel attacks are techniques to extract keys or secret elements from cryptosystems otherwise unbreakable by cryptanalysis or brute force. The instant power dissipation of a device has been studied first because it corresponds to a practical scenario, especially for smartcards. Indeed, those embedded devices receive their power from the outside. A rogue reader can thus supply the card while recording the instant current drawn, typically with a fast acquisition card. Based on these measurements, so-called differential (DPA [248]) or correlation power analyses (CPA [60]), referred to as in the sequel by the same generic term "DPA", can be mounted. DPA exploits the coincidence of two properties that characterize every cryptographic algorithm. On the one hand, it is always possible to exhibit an internal variable dependent on a manageable subset (*i.e.* small, usually 6 or 8 bits) of the key and of the input or output data. On the other hand, in "high threshold voltage" technologies, CMOS gates consume only when toggling. Therefore the power consumption is directly proportional to the circuit's activity. The power consumption due to the internal variable activity can be extracted from the circuit power traces by correlation with a power model. The attacker makes guesses about the unknown key subset and for each of them computes the correlation function. The larger correlation will betray the correct key hypothesis. Any unprotected

cryptographic implementation is thus vulnerable to DPA, because any use of the key bits leads to an information leakage in the power dissipation.

One way to protect a device from the DPA is to make its power consumption independent from the input data and key, by making it constant. This is the aim of the dual-rail with precharge logic (DPL [97]). This logic ensures by design a constant toggling rate irrespective of the data manipulated.

In dual-rail, every Boolean variable a is represented by a couple of two wires (a_0, a_1); when a is valid, $a = 0 \Leftrightarrow (a_0, a_1) = (1, 0)$ and $a = 1 \Leftrightarrow (a_0, a_1) = (0, 1)$. The convention to signal that A is not valid is $a_0 = a_1$. Every computation consists in one precharge (where a is invalid) followed by one evaluation (where a is valid). a_0 and a_1 are complementary. Whatever the input and key, exactly one and only one of (a_0, a_1) will toggle. The number of toggles is thus constant, so should be the overall power consumption.

Wave dynamic differential logic (WDDL [456]) and SecLib (Secured Library [189]) are two DPL solutions. WDDL is a DPL logic that makes use of the standard cell library. SecLib is another DPL logic that relies on customized balanced cells set that furthermore synchronize their inputs before evaluating.

In addition to comparing insecure logics with WDDL and SecLib, a second goal of this paper is to study further refinements of DPL logics consisting in balancing the layout. WDDL instances are separable: each gate is made up of two independent halves. Therefore, the two dual instances can be designed to have the same structure. They can also be constrained to be placed side-by-side. All DPL logics can be forced to have a balanced interconnection between them, and, on top of that, the wiring can be shielded. In MDPL [359], it has been suggested an alternative method to balance a netlist with a *deus ex machina* mechanism, consisting in randomly swapping the signification of a_0 and a_1 according to one single-bit mask. However, this protection can be defeated easily by a so-called PDF-attack [405]. This attack becomes all the more difficult as the netlist is already well balanced without any masking. This is the main focus for our work.

In order to compare logics styles and backend countermeasures, a chip called `SubBytes` has been realized. It embeds thirteen versions of the "SubBytes" function, the substitution box used in the AES algorithm [337]. Its purpose is to enable a comparative evaluation of the several implementations of the same combinatorial block.

The rest of the article is organized as follows. Section E.2 presents the security features that are implemented in the `SubBytes` circuit. The section E.3 gives conclusions about the expected security level using a static evaluation based on the layout study. Next, section E.4 is dedicated to the dynamic evaluation based on actual experiments. In this section, the specifications of the ASIC floorplan, programming model and drivers are described and motivated; then, an experimental evaluation of each SubBytes module is carried out. The section E.5 is a discussion about the relevance of design-time security metrics, the efficiency of WDDL *versus* SecLib, and the usefulness of backend-level counter-measures. Finally, section E.6 draws the conclusions of the paper and opens further research perspectives.

E.2 Presentation of the Security Features Embedded into the SubBytes Chip

E.2.1 Thirteen versions of the AES SubBytes Combinatorial Function

For the realization of the SubBytes chip, four libraries of cells were assessed:

1. Standard cell (CORE9GPLL library from STMicroelectronics, version 4.1),
2. Read-Only Memory ("ROM", generated by the STMicroelectronics Unicad tool – ugnLib),
3. WDDL [456] and enhanced-WDDL, based upon CORE9GPLL,
4. SecLib [193], a custom secure quasi-delay independent (QDI) logic. SecLib is similar to the quasi-delay insensitive logic presented in [317]: its structure is however optimized and it does not support the errors reporting capability.

The two first libraries (standard cells and ROM) are unprotected, and can thus constitute references for the security evaluation. The SubBytes chip embeds four unprotected instances with the following architectures:

1. **Standard cell**, described in VHDL as look-up table [337, p. 16] (called stdcell_lut),
2. **Standard cell**, factored in $GF(16)^2$, as suggested by Vincent Rijmen [382, 388, 475] (called stdcell_gf),
3. **Standard cell**, in a decode/permute/encode architecture presented by Guido Bertoni [27], depicted in Fig. E.1 (called stdcell_gb) [1],
4. **Layout-level generated** low-power contact-programmable ROM.

The references [449] and [450] are comprehensive studies of the different hardware architectures of the AES SBox and will help the reader in having a complete overview of the SBox in any dimension: security, area and power dissipation.

The secured implementations, WDDL and SecLib, embedded in SubBytes both resort to DPL. The SecLib cells are part of a full-custom library [189]. The WDDL and SecLib gates we consider in the sequel contain only one- and two-input gates.

WDDL, illustrated on the example of an AND gate in Fig. E.2, suffers from two identified weaknesses:

1. The two dual standard cells making up the WDDL gate are structurally different. They thus consume slightly different amounts of power with different current signature.
2. Depending on the arrival order of its inputs, the gate activity occurs at different instants [439].

The first issue can be fixed, employing what we call "enhanced-WDDL" (or eWDDL for short) cells, based on the 3-input majority standard cell from the STMicroelectronics

1. The work of Guido Bertoni *et al.* has been extended by Matteo Giaconia *et al.* in [138]. It yields an even more balanced design and thus achieves a still higher DPA resistance. However, at the date of the tape-out of the SubBytes circuit, we were not aware of this implementation, which explains it is not included into the ASIC.

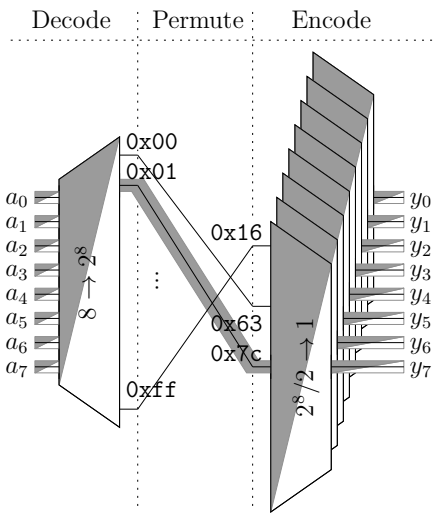


Figure E.1: The decode/permute/encode low-power and unprotected architecture for SubBytes.

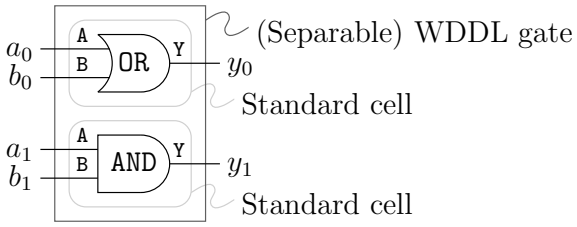


Figure E.2: The WDDL "AND" functionality.

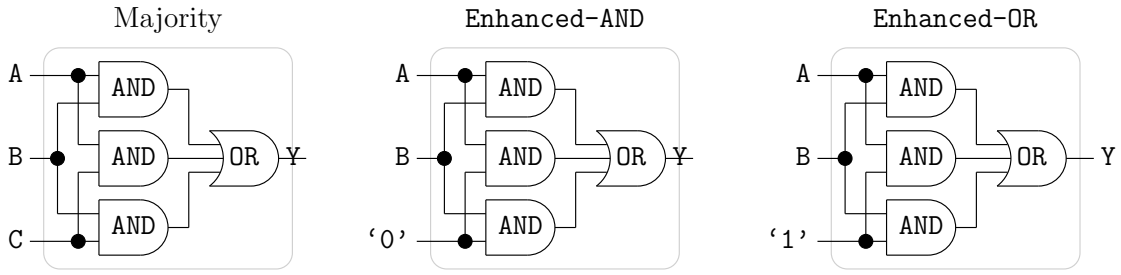


Figure E.3: The majority standard cell, called `AO5NLL` in STMicroelectronics library, can be specialized as two enhanced-WDDL cells implementing the *true* "`AND`" and "`OR`" functionalities.

library: $(\mathtt{A},\mathtt{B},\mathtt{C}) \mapsto \mathtt{A}\cdot\mathtt{B}+\mathtt{B}\cdot\mathtt{C}+\mathtt{C}\cdot\mathtt{A}$. The schematic of the majority and of the two enhanced WDDL derived cells are given in Fig. E.3. Those cells use the same architecture as MDPL [359], albeit with a constant hardwired mask.

The second issue of WDDL cannot be solved at the implementation-level, because it is fundamentally logical.

The figure E.4 shows that both issues are definitely fixed in SecLib:

1. All evaluations activate the same number of indiscernible logic gates: one `C`-element [414] and two `OR` gates. This contrasts with WDDL with which either a `AND` or a `OR` gate is activated.
2. The head `C`-elements synchronize the signals, thus preventing the gate from evaluating early.

The logic underlined in gray in Fig. E.4 is activated in the transition from precharge to evaluation and vice-versa.

The implementation-level variations amongst the secured cells are many-fold. They are defined and described in the list below:

$\mathcal{B}1$: identity of the dual gates,

$\mathcal{B}2$: differential placement,

$\mathcal{B}3$: differential routing,

$\mathcal{B}4$: differential dummies,

$\mathcal{B}5$: shield by global wires of each dual pair of wires,

$\mathcal{B}6$: complete module area shield by a top-level metal coating plane.

The first backend feature $\mathcal{B}1$ makes the computational paths to the true and the false outputs indistinguishable. The second item $\mathcal{B}2$ requires the gate to be somehow separable into two halves (refer to [191, appendix A]). Differential placement lessens the risks of unbalancedness due to variations from one location to another across the circuit's die. This helps reduce the disparities in power consumption, but most importantly, the constraint placement tends to make the routing (automatic, *i.e.* not constrained) similar

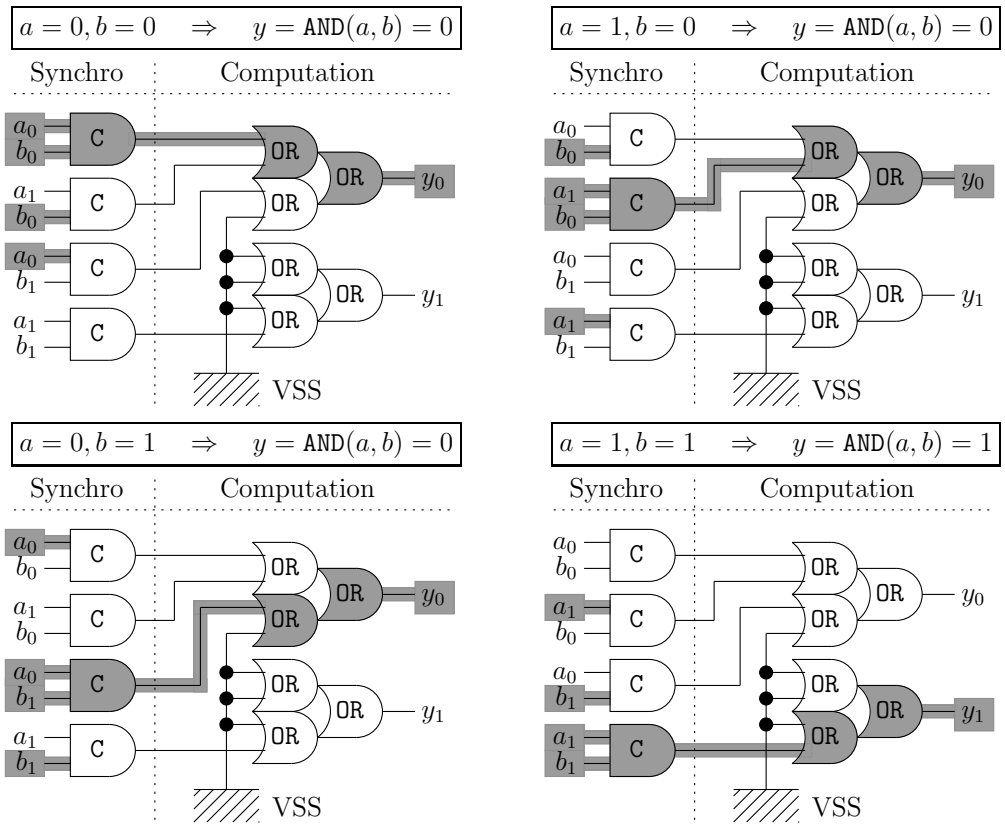


Figure E.4: The SecLib `AND` function activates always the same number of gates (Computation part) and is guaranteed to be immune from early evaluation thanks to the synchronizing `C`-elements.

for each gate. This pseudo-differential routing has the beneficial consequence that the load of each gate dual outputs is sensibly the same, so does the power consumption. The third item $\mathcal{B}3$ depends on the second one: differential routing can only be achieved provided the placement is also differential. It ensures a perfectly parallel hence balanced routing. It is performed thanks to the "backend-duplication" method [191]. The fourth item $\mathcal{B}4$ depends in turn on the third one: dummy metal slots can be spread in a differential way only if the routing is differential too. We recall that dummy slots are non-functional pieces of metal scattered "randomly" for the layer to reach a minimal density. This is indeed specified by the design rules manual in order to guarantee the planarity after chemical-mechanical polishing during fabrication. A constant planarity guarantees that the differential wires keep the same thickness along their route. The fifth item $\mathcal{B}5$ protects every dual pair of wires from cross-talk by placing a global (`vss` or `vdd`) wire between them. Usually, only the ground `vss` is used for shielding because it collects and drains all the parasitic currents without injecting the noise of the supply `vdd` into the integrated circuit's core. However, combining `vss` and `vdd` is an option worth considering when the power is not noisy because the shield wires can serve as a pervasive supplying network. They thus keep the voltage drops of the underneath logic cells as low as possible [188]. The sixth item $\mathcal{B}6$ is independent from the others and consists in coating the SubBytes module with a top-level metal (M6) plane. This shield against electromagnetic analyses (EMA [134]) is not studied in this paper, because we focus exclusively on power analysis.

Thirteen SubBytes modules are designed, combining the various logic styles and implementation-level options. They are detailed in Tab. E.1. In the sequel they are either referred to by their number or by their nickname, given in first and second columns respectively.

The unprotected implementations (modules (1) to (4)) do not benefit from any differential feature ($\mathcal{B}1$ to $\mathcal{B}5$), hence the "not applicable" (abbreviated "n/a") indications in the table.

Not surprisingly, secured implementations (modules (5) to (13)) suffer from a large overhead in terms of silicon area. Thus, the most compact architecture amongst the unprotected (namely (1), *aka* `stdcell_gf`) is selected as a reference. The performances of the thirteen modules are given in Tab. E.2. This table clearly shows that the synthesizer tries hard to use as many cells as possible from the library for the straightforward LuT architecture (53 unique instances as for (2)), to the detriment of a global optimization (such as the smaller implementation (1), that uses only 22 unique instances).

E.2.2 Projected Security Level of DPL Versions of SubBytes

The security level of WDDL and SecLib (with the same security features as `wddl_4` and `seclib_4`) has already been studied in simulation in [188], and from experimental measurements done *in silico* in [175]. In contrast, this article explores a trade-off between security features and cost overhead. Thus, we investigate degraded (*i.e.* sub-optimal) backend-level countermeasures with respect to WDDL and SecLib.

The expected security partial order is expressed by the "$\prec$" operator in Fig. E.5.

Table E.1: Security features $\mathcal{B}1$ to $\mathcal{B}6$ of the thirteen SubBytes modules.

#	Nickname	$\mathcal{B}1$ (Gate)	$\mathcal{B}2$ (Placement)	$\mathcal{B}3$ (Routing)	$\mathcal{B}4$ (Dummy)	$\mathcal{B}5$ (Shield)	$\mathcal{B}6$ (EMA)
(1)	stdcell_gf	n/a	n/a	n/a	n/a	n/a	no
(2)	stdcell_lut	n/a	n/a	n/a	n/a	n/a	no
(3)	stdcell_gb	n/a	n/a	n/a	n/a	n/a	no
(4)	rom	n/a	n/a	n/a	n/a	n/a	no
(5)	wddl_0	no	no	no	no	no	no
(6)	wddl_1	no	yes	no	no	no	no
(7)	wddl_2	no	yes	yes	yes	no	no
(8)	wddl_4	no	yes	yes	yes	yes	no
(9)	ewddl_4	yes	yes	yes	yes	yes	no
(10)	seclib_1	yes	yes	no	no	no	no
(11)	seclib_2	yes	yes	yes	yes	no	no
(12)	seclib_4	yes	yes	yes	yes	yes	no
(13)	seclib_4ema	yes	yes	yes	yes	yes	yes

Table E.2: SubBytes blocks physical characteristics.

#	Area [μm^2]	#! instances	# instances	Density
(1)	1 767	22	144	98.6 %
(2)	4 018	53	423	98.1 %
(3)	4 841	53	548	98.6 %
(4)	12 830	n/a	n/a	n/a
(5)	8 981	2	342 × 2	95.8 %
(6)	10 760	3	449 × 2	93.7 %
(7)	10 844	3	449 × 2	93.0 %
(8)	16 097	3	449 × 2	62.5 %
(9)	16 944	3	451 × 2	75.9 %
(10)	23 468	8	166	88.2 %
(11)	25 586	8	166	80.9 %
(12)	25 417	8	166	81.4 %
(13)	25 417	8	166	81.4 %

$$\left[\begin{array}{ll} (1) = (2) = (3) = (4) \prec \left\{\begin{array}{l} (5) \prec (6) \prec (7) \prec (8) \prec (9) \\ (10) \prec (11) \prec (12) = (13) \end{array}\right. & \text{// See note 1.} \\ (6) \prec (10) & \text{// See note 2.} \\ (7) \prec (11) & \text{// See note 2.} \\ (8) \prec (9) \prec (12) & \text{// See note 2.} \end{array}\right.$$

Figure E.5: Expected security order of the 13 modules embedded into the ASIC `SubBytes`.

Note 1 Unprotected implementations have (*a priori*) a comparable level of security (no counter-measure.) Secured libraries, based upon either WDDL or SecLib, are expected to be more secure. Differential placement, differential routing and differential dummies are counter-measures that are built on top one of each other to increase the security provided by the cell library. EMA [134] shield is not expected to impact the protection against the DPA [248].

Note 2 Everything being otherwise comparable (differential placement, routing and metal dummies), WDDL is expected to be weaker than SecLib [188, 175]. The reason is that at the "silicon"-level, SecLib is more balanced than WDDL. Further "metal"-level (*i.e.* interconnect) security features will enhance the security, but will most probably not make up for the "silicon"-level (*i.e.* logic) discrepancies.

E.2.3 Evaluation Methodology for the Simulations & the Experimental Measurements

There are two ways to evaluate the security level of the competing SubBytes modules.

1. *Static* evaluation considers the layout and tries to find dissymmetries in it. Statistics on the nets can be collected from the netlist. The dispersion of the characteristics across nets is considered a measurement of static unbalancedness. This evaluation strategy is called "design-time" because it does not require to have a silicon prototype at disposal. This is the approach carried on in Sec. E.3.
2. *Dynamic* evaluation considers the global behavior of each SubBytes module. Statistics are realized by trying all possible input configurations. The approach is thus either a simulation or real-world measurements on a silicon chip. The latter requires a device, and thus costs more because the whole fabrication process must be realized. However, it is also a more accurate than simulation because it places the evaluator in the same shoes as a potential attacker. Section E.4 concentrates on this aspect of the evaluation.

E.2.4 Motivation for Combinatorial Gates Study

Most side-channel attacks are based on correlations with an intermediate variable. In both software and hardware implementations, a variable is stored in a register. From an attacker's standpoint, this is a great opportunity since the register activity is reproducible

in time. This means that statistics will coherently correlate with either the register contents or its contents change.

It thus appears that combinatorial logic is seldom studied as an exploitable source of leakage. The main reason is probably that the relationship between the activity of this logic and the data it computes is far from being obvious: combinatorial gates evaluate at data-dependent dates and might even produce non-functional transitions (called glitches) whose impact on the power dissipation is difficult to model. Moreover, while the number and exact location of the registers are quite simple to guess or reverse-engineer, the structure of combinatorial logic is much more difficult to figure out. So, paradoxically, although in some algorithms such as AES the combinatorial logic makes up about 80 % of the implementation area and power dissipation, it happens not to be the most frequent target for a side-channel attack.

One example where the analysis of a combinatorial net has been successful was the attack of a masked sbox, by Stefan Mangard *et al.* at CHES'05 [294]. Amongst the whole netlist, they identified the net that was the less dependent in the mask, and focused the attack on the variable it carried. However, apart from this very special situation, inner nets within combinatorial logic are not the most frequently encountered candidates for a side-channel attack.

But in a circuit where the registers are perfectly protected, the only remaining sources of data dependency are the Boolean logic gates. This is the assumption we made in this article and the reason why we focused on combinatorial parts evaluation.

Figure. E.6 illustrates this. It shows the voltage drop over a spy resistor monitoring the instant current consumed by an unprotected DES module during two clock cycles. The registers consume current at the clock rising and falling edges of the clock, whereas the combinatorial logic consumes current only after the registers have evaluated, typically a couple of nanoseconds after the rising edge of the clock. It seems easy to balance the registers, because there are not so many of them in an implementation. However, the combinatorial parts are numerous and complex. Both DPA and template attacks could target the variations in the combinatorial parts even if the registers are exactly balanced.

The sbox, for instance, offers room for concrete attack thanks to its mathematical properties. If we denote by S the functionality of the sbox, then a correlation attack basically consists in evaluating an auto-correlation of S (between the measurements and the guessed model). When the correct key is guessed, the auto-correlation is maximal, equal to $(S \otimes S)(0)$. Otherwise, the correlation yields $(S \otimes S)(\epsilon) \leq (S \otimes S)(0)$, where ϵ is the error on the key guess. In case the exclusive-or operation is used to mix the key with the datapath, $\epsilon \doteq k_{\text{actual}} \oplus k_{\text{guessed}}$. The contrast of an auto-correlation is all the higher as the sbox S is non-linear [194, 362, 64, 196]. For cryptanalytic reasons, the sboxes are chosen as highly non-linear. In this respect, the abstract function of S, rather than its implementation, helps the attacker in her decision for the correct key: mathematical properties of S allow to discriminate efficiently the different key candidates.

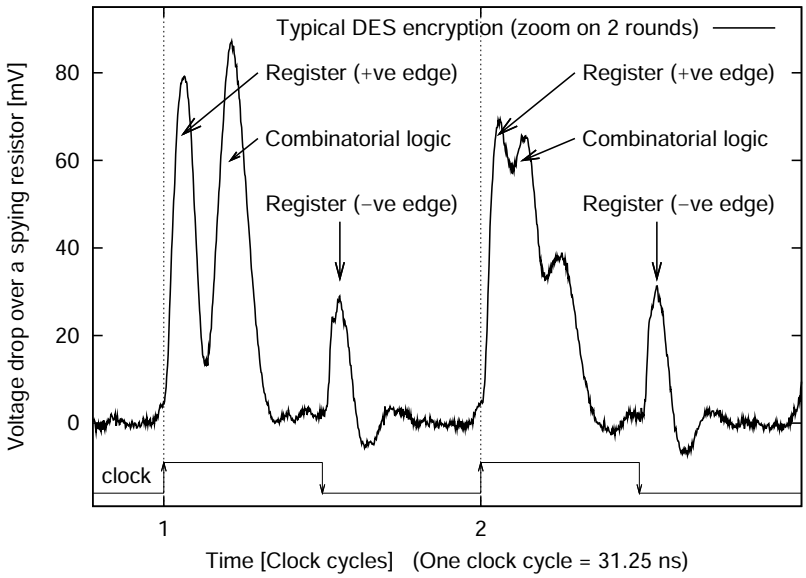


Figure E.6: Typical voltage trace of an unprotected DES module. This quantity is proportional to the instant current consumed by the chip. The sequential (positive & negative clock edges) and combinatorial currents are identified by arrows.

E.3 Static Evaluation of the Security of Nine `SubBytes` Dual-Rail Modules

The security of the dual-rail modules is assessed statically based on the study of differential routing unbalancedness. In order to reach this goal, the resistance "R" and capacitance "C" for every net of the dual-rail modules have been extracted after completion of all backend steps (*i.e.* placement, routing, dummies insertion). The extraction tool is `rcOut`, provided with Cadence software suite SoC/Encounter version 6.1. Without any surprise, we observe that all the resistances match pair-wise, because this quantity depends only on the geometry of the nets. In contrast, the capacitances are different from one regular net to its dual, since capacitances are cross-coupled with the neighboring nets, and the neighborhood of each dual net differs. For each net, the relationship between the parameter "C" extracted from the layout and the instant current drawn by the driver when it switches is linear: $I = VDD \times C$, where VDD is the nominal power supply voltage measured relatively to the ground. Therefore, the ratios between true and false nets capacitances, denoted C_1 and C_0, are computed. Any deviation from 1 is a dissymmetry. Indeed, the observable side-channel amplitude is $|I_1 - I_0| = VDD \times |C_1 - C_0|$, which is non-zero if and only if (iff) $C_1/C_0 \neq 1$. The logarithm of these quantities is plotted in Fig. E.7 and E.8 to allow for a duality-wise agnosticism:

– if the load of a true net is ε more than its false counterpart, then $\log(\frac{1+\varepsilon}{1}) \approx$

$\underline{\boxed{+\varepsilon}} + \mathcal{O}(\varepsilon)$, whereas

– if the unbalancedness is the opposite, $\log(\frac{1}{1+\varepsilon}) \approx \underline{\boxed{-\varepsilon}} + \mathcal{O}(\varepsilon)$, which is "fair" w.r.t. the true/false duality: the penalty is exactly the opposite at first order, hence the same in absolute value.

Fig. E.7 shows dispersion in the so-called "default mode", where capacitances are extracted only w.r.t. the ground (`vss` = 0 volt). In Fig. E.8, cross-capacitances between nets are extracted too, in a π-model, also called "detailed mode". Thanks to the usage of the logarithmic scale, the dispersion profiles are centered around 0. One can notice that they are more or less scattered. The dispersion is ideal in the "default mode". The values for module (12), for instance, present the shape of a Dirac peak with the chosen quantification of 1 %. The "detailed mode" better captures the unbalancedness due to the neighborhood dissymmetry: the same module (12) does show an appreciable dispersion in C_1/C_0. The module (13) is not represented because its coupling with the ground is strictly equal to that of (12).

In order to easily compare these dispersions, we compute the standard deviation, also abbreviated "std_dev" in the sequel. Those figures are given in Tab. E.3. The module `wddl_0`, that is neither placed nor routed differentially, is — by far — the worst. For the other modules we need to notice that the nets capacitance is made up of two components:

1. The *wire* capacitance C_{wire}. The differential routing, the dummies and the shield are supposed to reduce the dispersion in the wire capacitance.
2. The *gate* input capacitance C_{gate}. The logic style is expected to impact this part of the capacitance dissymmetry: WDDL is not balanced in the gates inputs, because the dual gates are different, whereas eWDDL and SecLib logic are.

The average ratio between the wire capacitance and the total capacitance $C_{total} \doteq C_{wire} + C_{gate}$ is about 50 %, which means that dissymmetries in *wires* and *gate* inputs are to be fought with the same amount of efforts. Behind `wddl_0`, the WDDL modules `wddl_{1,2,4}` come next, due to the unbalancedness of the gates input capacitances. The dispersion of C_{gate} (of inputs `A` and `B`) in WDDL can be observed in histograms (6), (7) & (8) of Fig. E.7. Apart from the inverter cells, their netlists are made up exclusively of `AND` and `OR` standard cells, that happen to have different input capacitances:

$$\log\left(\frac{C_{\texttt{AND:A}}}{C_{\texttt{OR:A}}}\right) = \log\left(\frac{1.63\ \text{pF}}{1.53\ \text{pF}}\right) = 0.063 \quad \approx$$
$$\log\left(\frac{C_{\texttt{AND:B}}}{C_{\texttt{OR:B}}}\right) = \log\left(\frac{1.43\ \text{pF}}{1.34\ \text{pF}}\right) = 0.065\,.$$

The other modules (eWDDL and SecLib) have the input gates balanced, and thus feature a smaller dispersion, because only the routing dissymmetry remains. For both WDDL and SecLib, it is clear that the back-end duplication does help (`wddl_1` *vs* `wddl_2` and `seclib_1` *vs* `seclib_2`). Notice that in Fig. E.7, SubBytes module `seclib_1` (10) seams at first glance to be more dispersive than module `seclib_2` (11). However, some rare net couples, with $|\log(C(\text{true})/C(\text{false}))| \approx 0.2$, are very unbalanced in module `seclib_2`, which explains the results obtained in Fig. E.7 default mode:

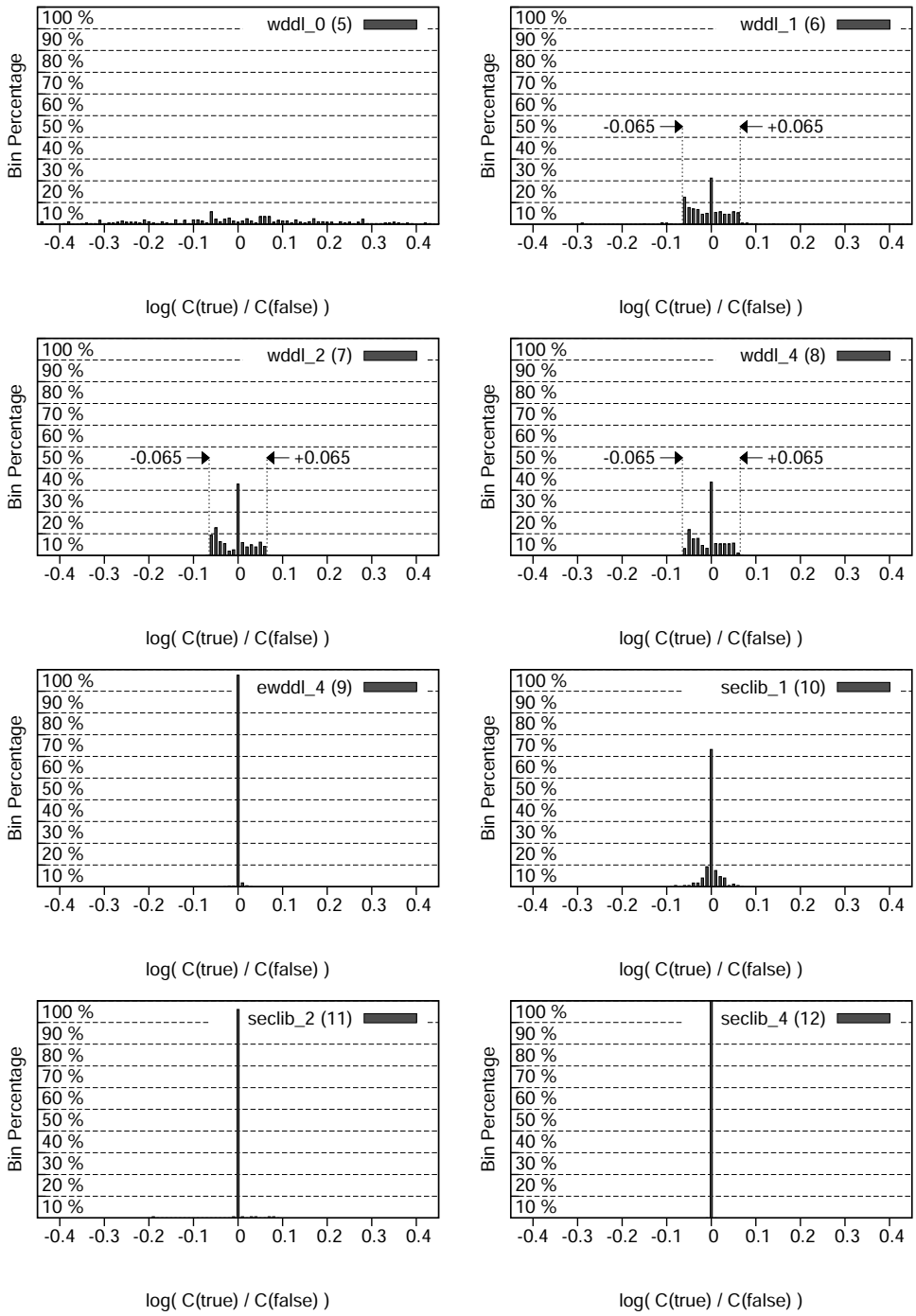


Figure E.7: Distribution of the extracted deviation from the perfectly balanced dual-rail pair (*default extraction mode.*)

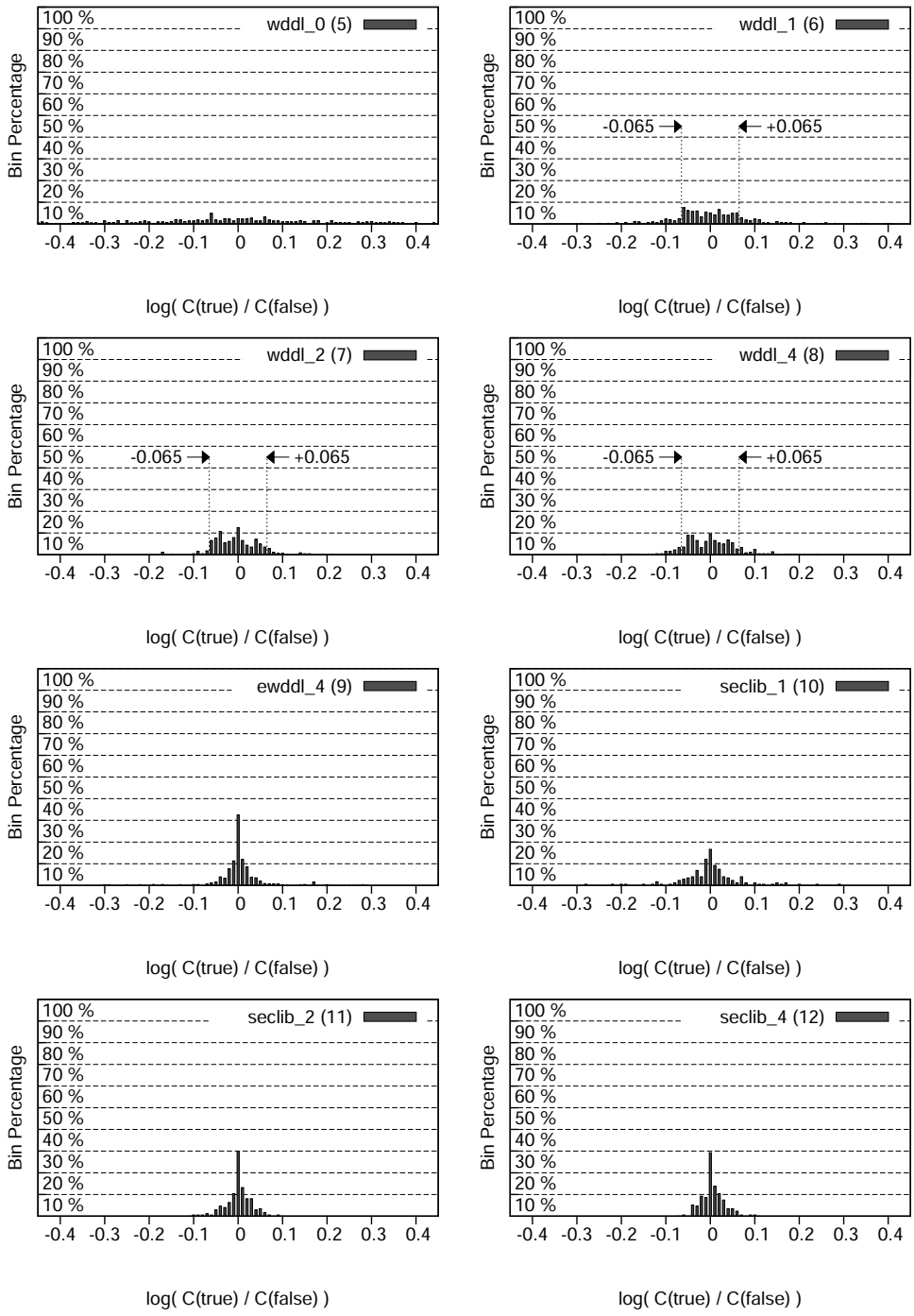


Figure E.8: Distribution of the extracted deviation from the perfectly balanced dual-rail pair (*detailed extraction mode.*)

Table E.3: SubBytes dual-rail blocks capacitive dispersion, computed from the statistics collected in Fig. E.7 and E.8.

#	Nickname	Std_dev (default mode)	Std_dev (detailed mode)	$\frac{C_{wire}}{C_{total}}$
(5)	wddl_0	68.58 $\times 10^{-3}$	77.71 $\times 10^{-3}$	55 %
(6)	wddl_1	2.73 $\times 10^{-3}$	7.62 $\times 10^{-3}$	65 %
(7)	wddl_2	1.21 $\times 10^{-3}$	2.67 $\times 10^{-3}$	68 %
(8)	wddl_4	0.94 $\times 10^{-3}$	3.56 $\times 10^{-3}$	70 %
(9)	ewddl_4	0.00 $\times 10^{-3}$	2.44 $\times 10^{-3}$	52 %
(10)	seclib_1	0.26 $\times 10^{-3}$	4.95 $\times 10^{-3}$	52 %
(11)	seclib_2	0.30 $\times 10^{-3}$	0.81 $\times 10^{-3}$	57 %
(12)	seclib_4	0.00 $\times 10^{-3}$	0.62 $\times 10^{-3}$	55 %

std_dev(11) > std_dev(10). Anyway, we recall that only the detailed mode provides a sufficiently accurate estimation of the nets average unbalancedness, hence of the layout static security.

A similar analysis as the one of Sec. E.2.2 is carried out in detailed extraction mode, regarding only static evaluators for the routing. The expected level of security is depicted in Fig. E.9. This figure shows that the security level of competing designs can be predicted using methods inspired from the two-dimensional chromatography. Notice that, compared to the overall security expectation (taking into account both the *logic gates* and their *interconnect*) discussed in Sec. E.2.2, a new relationship is established: (9) is assumed to be of equal quality as (12), because:

- eWDDL and SecLib have balanced C_{gate} (security feature $\mathcal{B}1$), and
- their interconnect is balanced with the same differential features $\mathcal{B}2$ to $\mathcal{B}5$.

If we compare this figure (Fig. E.9) and the statistical results obtained in Tab. E.3, it appears that the predictions are all valid, but for the effect of the pairs shielding. Indeed, we have predicted (7) $\prec$ (8) and (11) $\prec$ (12) = (9), but we have neither std_dev(7) > std_dev(8) nor std_dev(11) > std_dev(9). The reason might be that the SubBytes modules are too small for the metal lines to have the opportunity to be cross-coupled. The effect of the shield is merely to increase globally the routing length, and thus paradoxically to increase unequally the capacitive parasitics. This agrees with this intuitive observation on the larger modules (11) & (12): they do satisfy (11) $\prec$ (12). Therefore, the two violations (7) $\not\prec$ (8) and (11) $\not\prec$ (9) can safely be considered artifacts that do not scale up for a complete algorithm protection, with many substitution boxes and a complex datapath.

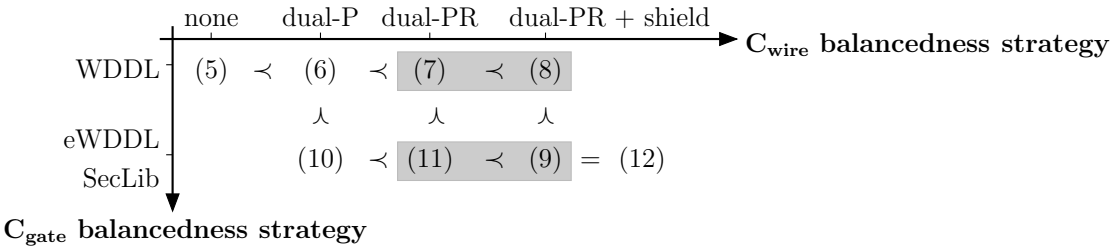


Figure E.9: Expected level of security partial order, based on the sole static criterion. The gray boxes indicate security relationships that are violated by the extraction in "detailed mode" statistics.

E.4 Experimental Comparison of the Thirteen `SubBytes` Modules

E.4.1 Implementation into a Single-Chip Prototyping ASIC

The thirteen SubBytes modules studied in the previous section have been implemented in an ASIC. Their position on the floorplan is indicated in Fig. E.10. There are only four functional I/O pads, common to all SubBytes modules: this way, they are all evaluated under the same experimental conditions. The I/O pads are:

1. `clk`: a global clock to synchronize the executions,
2. `data_in`: an input serial line,
3. `data_out`: an output serial line,
4. `enable`: a selection signal deciding whether the circuits loads bits serially from the outside or transfer them in parallel into the substitution boxes.

To reduce noise, pads, core and SubBytes modules are powered from three different sources, all operating under a nominal 1.2 V voltage. The list of all the pads is given in Fig. E.11.

The pictures of the ASIC and of its DIL48 (Dual In-Line package with 48 pins) cavity are given in Fig. E.12.

E.4.2 `SubBytes` Programming Model

For the thirteen SubBytes modules to be operated in a unified way, they require a common programming paradigm. The chip architecture is based on a shift-register for serial registers load and flush. Thanks to a two-stage pipeline at the input and one-stage pipeline at the output of the SubBytes blocks, the data are presented in front of all SubBytes modules. To suppress the power consumption of one specific module we freeze its inputs. For example, it can be always loaded the same data, say `0x00`. As a result, the circuit simply comprises $3 \times \mathtt{n}$ flip-flops (DFFs), where `n` is the total number of inputs

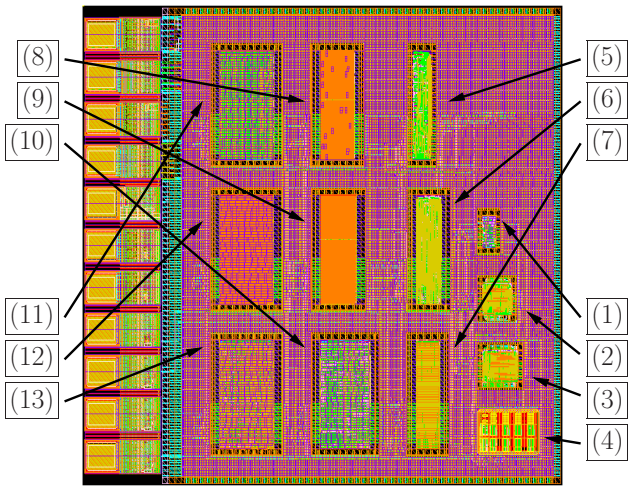


Figure E.10: The `SubBytes` circuit's layout.

	Pad name	Nature	#
	VDD_SUBBYTES_1V2	vdd1V2	31
	VSS_SUBBYTES_1V2	vss1V2	32
	VDD_CORE_1V2	vdd1V2	34
	VSS_CORE_1V2	vss1V2	35
	VSS_IOREF_CORE_1V2*	vss1V2	36
	VDD_PAD_3V3	vdd1V2	37
	VSS_PAD_3V3	vss1V2	38
Fonctional I/O pads	data_out	out	39
	data_in	in	40
	enable	in	41
	clk	in	42

**unconnected*

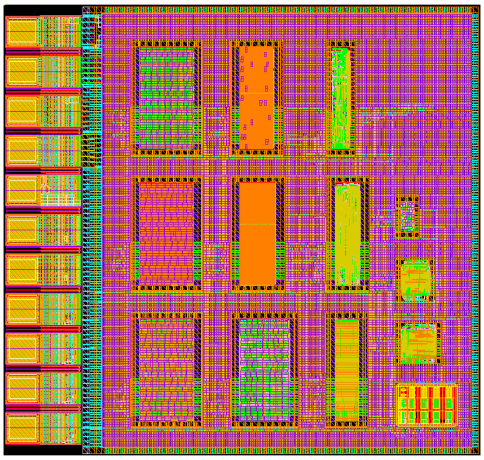


Figure E.11: Datasheet on the `SubBytes` circuit's pads.

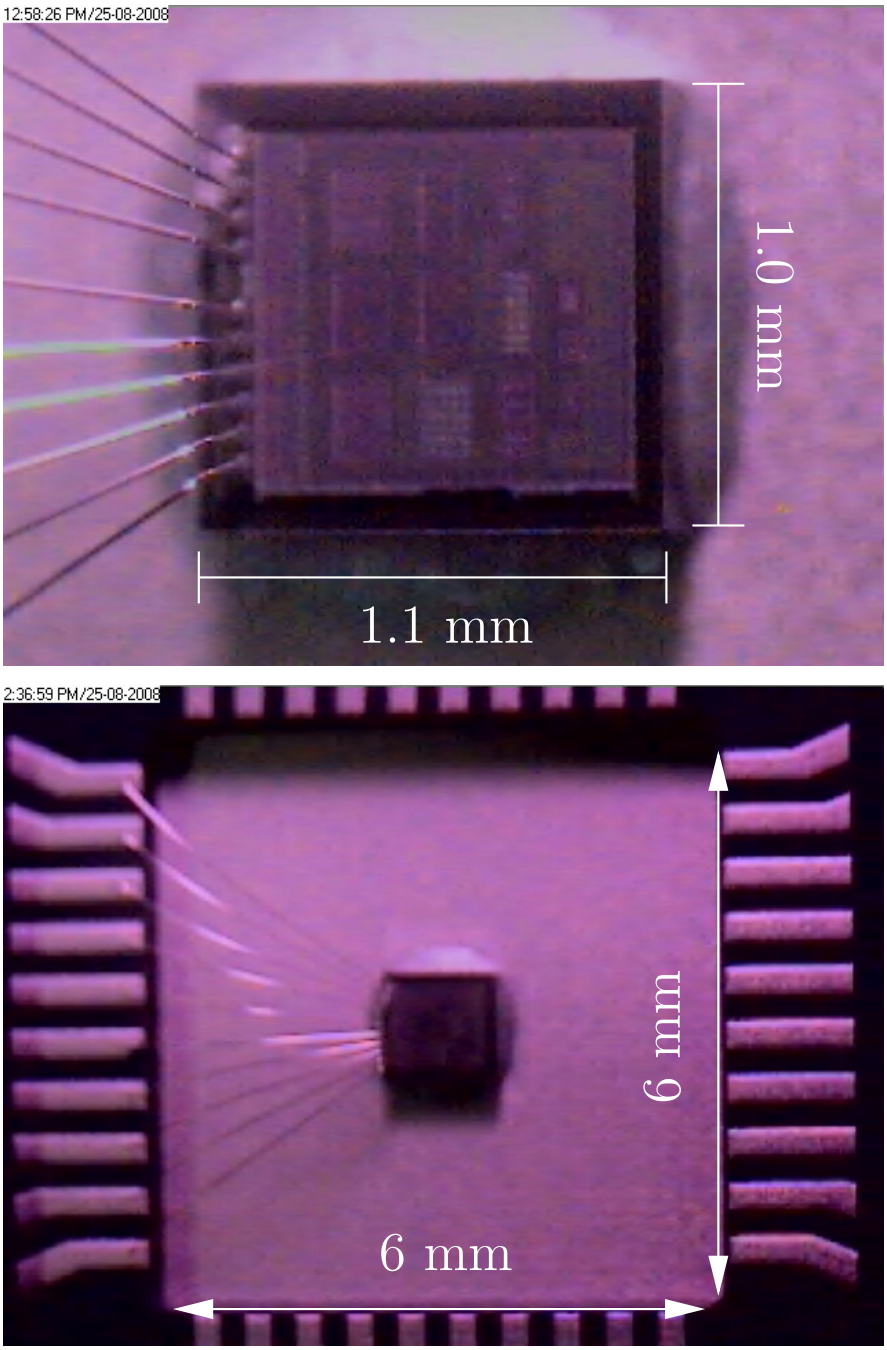


Figure E.12: The `SubBytes` circuit's monographs, as seen from an optical microscope.

of the combinatorial gates (as shown in Fig. E.13, `n` = 180 in `SubBytes`). The registers are divided into three `n`-bank registers, detailed below:

1. `reg_i1`: is a **parallelization** register for the input `data_in`,
2. `reg_i2`: is a register that performs the **transition** between two states *Initial* → *Final*,
3. `reg_o`: is a **serialization** register for the output `data_out`.

The registers behavior is described by the VHDL code snippet listed in Fig. E.14. The synthesis result is sketched in Fig. E.15.

The combinatorial logic is detailed in Fig. E.13. When `enable` is set to '0', `reg_i1` is used as shift-register. When it is set to '1', the `n` bits of `reg_i1` are transferred simultaneously into `reg_i2`.

More precisely, the two basic operations are:

1. "`shift`": sequentially load `reg_i1` with `n` bits $D_{i\in[0,\mathtt{n}[}$ sampled from `data_in`, and sequentially unload the `n` bits of `reg_o` on `data_out`,
2. "`transfer`": transfer `reg_i1` into `reg_i2` and `combi_out` into `reg_o`.

Notice that during `shift` (resp. `transfer`), `reg_i2` (resp. `reg_i1`) is left unchanged. In the `transfer` operation, the `data_in` input is *discarded*; in order to avoid confusion, it is thus safe to keep it *unchanged*. The control sequence to realize those operations is given in the waveforms shown in Fig. E.16.

WDDL and Seclib blocks have staggered registers, so as to keep the place-and-route dualization [191] even at the interfaces. This is shown in Fig. E.17.

Synthesis and place-and-route were performed with Cadence tools. The synthesizer is `bgx_shell` v05.15-s095+1, used with option `-BGX` for improved results on high-level behavioral VHDL [229] source code. The backend is realized by `First Encounter` v04.10-s415_1 and the interconnection routing by `NanoRoute` v04.10-s914. The chip was fabricated through the silicon broker CMP, that prepares the final layout and delegates the actual fabrication to STMicroelectronics' foundries.

The vertical routing direction has been chosen for M3 and M5 and the horizontal for M4 and M6.

As for the top-level metal M6 used to protect the circuit against EMA, it is actually not permitted to use it uniformly, due to stringent design rules about thermal stress. Instead, the so-called "metal-slot" design rules state that 9% of holes must be spread over the plane. The plane is thus a mesh obtained by the replication of the pattern depicted in Fig. E.18.

E.4.3 Experimental Environment

E.4.3.1 Enumeration of Required Power Traces Measurements for a Comprehensive Evaluation

The power measurements come down to testing the combinatorial functions exercised with all the possible transitions. For unprotected instances, the transitions consist in

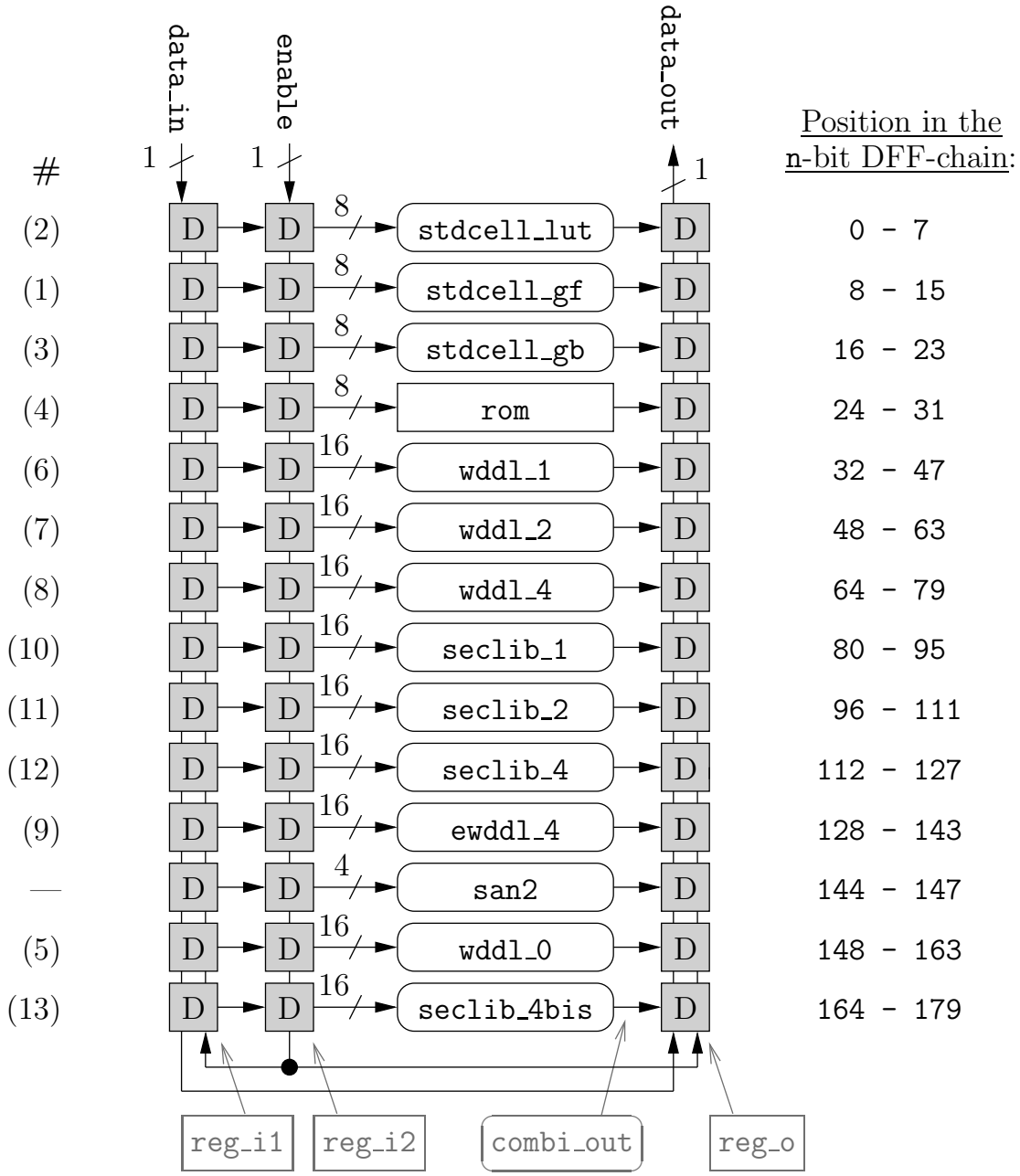


Figure E.13: The `SubBytes` circuit's sequential (*in gray*) and combinatorial (*in white*) architecture enabling a random programming. The module `san2` is not described since out of the scope of this article.

```
P_REG: process( clk ) begin
  if rising_edge( clk ) then
    if enable = '1' then -- Parallel transfer
      reg_i2 <= reg_i1;
      reg_o  <= combi_out;
    else                 -- The input/output registers are shifted
      reg_i1( n-1 downto 0 ) <= reg_i1( n-2 downto 0 ) & data_in;
      reg_o ( n-1 downto 0 ) <= reg_i1( n-1 ) & reg_o( n-1 downto 1 );
    end if;
  end if;
end process P_REG;

P_DATA_O: data_out <= reg_o( 0 );
```

Figure E.14: Behavioral description of the registers that parallelize and serialize the data for the combinatorial functions under test.

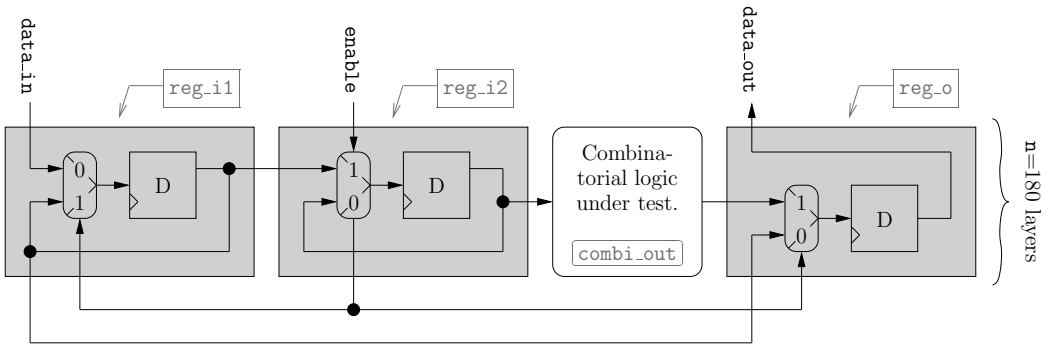


Figure E.15: The `SubBytes` circuit's sequential architecture. Refer to Fig. E.14 for a textual version.

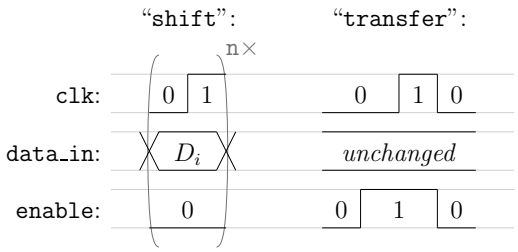


Figure E.16: The "`shift`" and "`transfer`" basic operations of the `SubBytes` circuit.

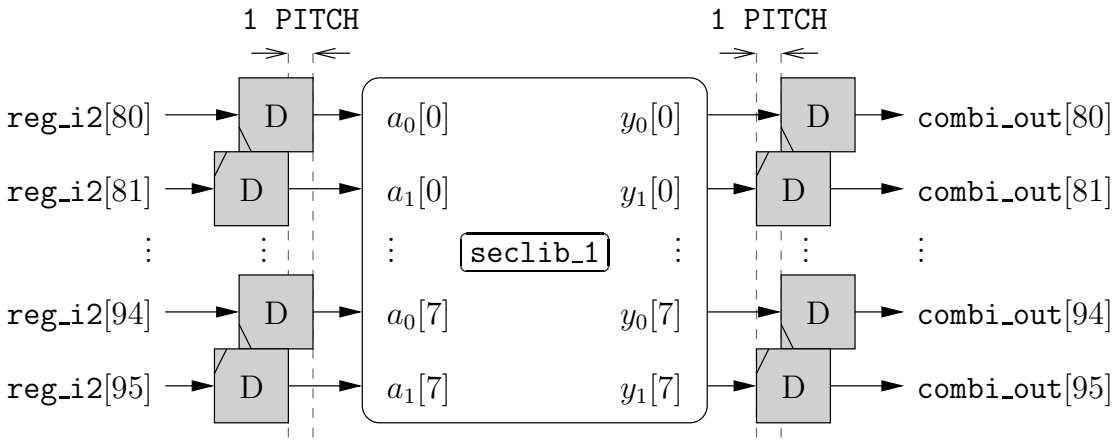


Figure E.17: Staggered register pairs at the interface of the dualized blocks (instances 5 to 13.)

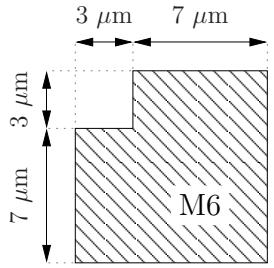


Figure E.18: M6 pattern for EMA-shield using a metal-plane mirror.

Table E.4: Number of distinct power measurements to realize on the SubBytes instances to fully characterize their signature.

Instance #	Transition count	Description
(1, 2, 3, 4)	$2^{2\times 8} = 65\,536$	$\forall i, f : i \to f$
(5, 6, 7, 8, 9)	$4 \times 2^8 = 1\,024$	$\forall i : 0 \to i, i \to 1, 1 \to i, i \to 0$
(10, 11, 12, 13)	$2 \times 2^8 = 512$	$\forall i : 0 \to i, i \to 0$

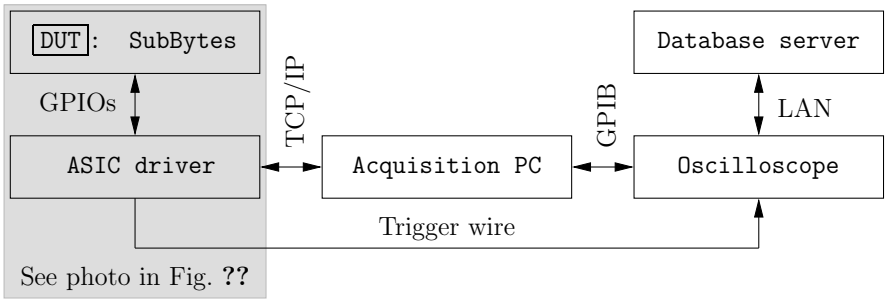


Figure E.19: Acquisition platform for `SubBytes` power traces.

changes from an initial value $i \in [0, 2^8[$ to a final value $f \in [0, 2^8[$. For secured instances, the protocol consists in transitions between a spacer and a valid state. The WDDL instances can be used both with the $\{00\}^8$ and the $\{11\}^8$ spacers, whereas only the null spacer $\{00\}^8$ is usable (unless making the gate insecure) for the SecLib-based instances. The number of measurements is summarized in Tab. E.4.

E.4.3.2 Acquisition Platform

The acquisition is managed by a central personal computer, that dialogues with:
- the device under test (DUT), namely the `SubBytes` ASIC, driven by an `ACME fox` (`http://www.acmesystems.it/`) development board, and
- a digital oscilloscope, in charge of acquiring traces and storing them in a `postGreSQL` database server.

The acquisition architecture is depicted in Fig. E.19. Two photographs of the *in-house* platform driving `SubBytes` are shown in Fig. E.20.

E.4.4 Experimental Evaluation Metrics

E.4.4.1 Definition of $M1$: Maximum Standard Deviation over a Complete Trace

To compare the diverse implementations, the following metric M_1 is used:

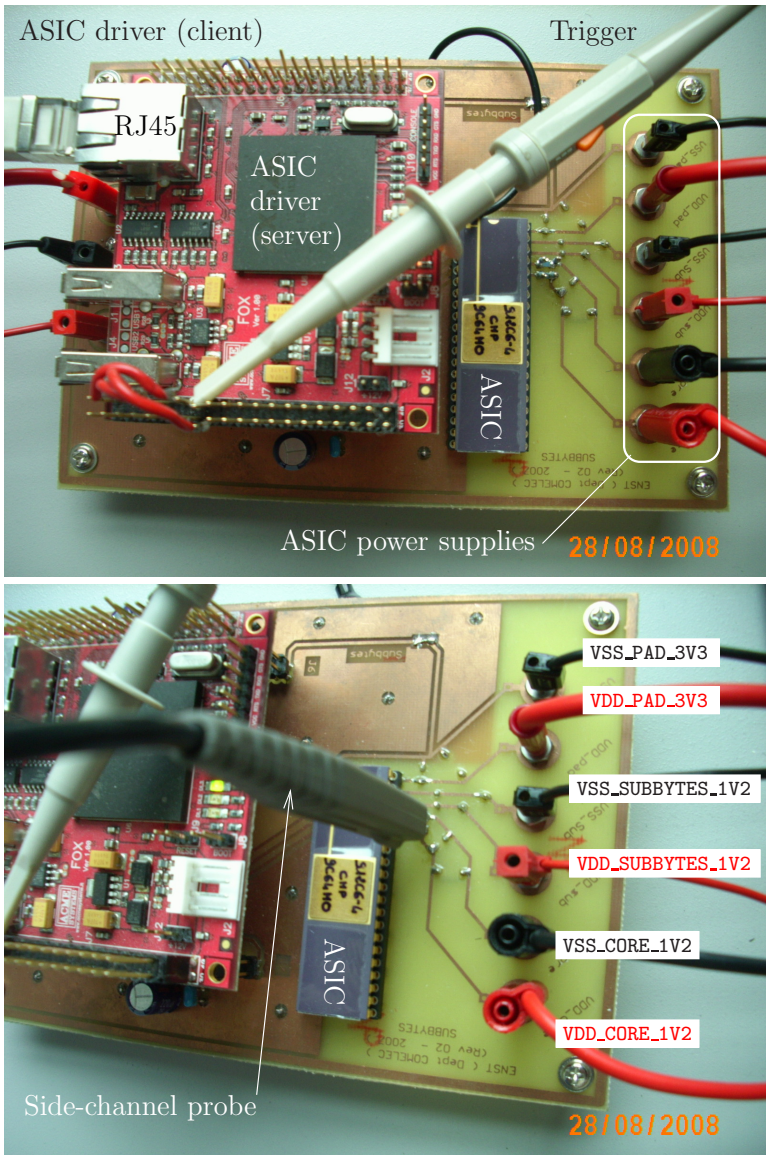


Figure E.20: Control board for `SubBytes` (ASIC under test) power traces.

– let $P(x \to y)(t)$ a power trace, acquired by the platform shown in Fig. E.20, with x and y in [0x00, 0xff] and t the time in one clock period $[0, T[$,
– let $P(t)$ be the average power trace over all the x (initial value) and y (final value),
– let $\sigma(t)$ be the traces standard deviation: $\sigma(t) \doteq \sqrt{\frac{1}{2^8 \times 2^8} \sum_{x,y} (P(x \to y)(t) - P(t))^2}$ (notice that $\sigma(t)$ is also a trace: it has as many points t as the any original trace),
– let M_1 be the maximum value taken by $\sigma(t)$ over all dates t.

M_1 focuses on the highest bias on a clock period, which makes sense in cryptographic applications where any singularity is exploited. It also concurs with the "mono-variate" bias one DPA will identify as the most leaking instant that correlates best with the leakage model.

Two other metrics, called M_2 and M_3, are also considered, as variations.

E.4.4.2 Definition of $M2$: Mean Standard Deviation over a Complete Trace

M_2 is the integral of the standard deviation over one clock period T, that is to say $\frac{1}{T} \int_{t=0}^{t=T} \sigma(t)\,\mathrm{d}t$. The metric M_1 is meant to be more stringent than M_2. However, $M2$ grasps variations over a full execution of SubBytes. It closely relates to "multi-variate" analyses, such as *templates* with a principal component analysis [13] where the principal direction is a step function over the evaluation clock period.

E.4.4.3 Definition of $M3$: Standard Deviation of an Averaged Trace

M_3 models a low-cost attack, where the attacker is supposed not to be equipped with a fast oscilloscope. The simulation of this scenario is obtained by first averaging the traces over one entire clock period, resulting in $P(x \to y) \doteq \frac{1}{T} \int_t P(x \to y)(t)\,\mathrm{d}t$. The metric M_3 is defined as the standard deviation of $P(x \to y)$.

E.4.4.4 Comparison and Analysis of Metrics

Table E.5 presents the three metrics calculated from these measurements. Due to a design error, the ROM (module number 4) is not fully functional (some addresses are unavailable). It is thus excluded from the table.

The single-ended modules (1), (2) & (3) are evaluated based on

1. one computation per clock cycle (65 536 averages) and
2. one computation every other clock cycle, with a precharge to zero in-between (256 averages).

It clearly appears that the single-ended modules operated with a throughput of one computation per clock cycle are much less secure than any dual-rail logic (5), (6), $\cdots$, (13). The gain of the dual-rail logic over classic CMOS logic is thus undebatable.

However, it is interesting to notice that some classic logics are affected by the sole use of a precharge. If we consider an interleaved precharge to 0x00, Tab. E.5 shows that:
– module (1), `stdcell_gf`, is not affected by the insertion of the spacer,
– module (2), `stdcell_lut`, becomes slightly more secure, whereas

Table E.5: Metrics for 12 implementations of `SubBytes`.

#	**Nickname**	10^3 **M1**	10^3 **M2**	10^3 **M3**
On 65 536 traces ($\forall i, f \in [\texttt{0x00}, \texttt{0xff}]^2 : i \to f$)**.**				
(1)	`stdcell_gf`	76.174	21.162	17.651
(2)	`stdcell_lut`	122.231	29.742	20.123
(3)	`stdcell_gb`	228.515	23.677	6.290
On 256 traces ($\forall f \in [\texttt{0x00}, \texttt{0xff}] : \texttt{0x00} \to f$)**.**				
(1)	`stdcell_gf`	83.903	21.828	19.488
(2)	`stdcell_lut`	82.038	21.838	17.644
(3)	`stdcell_gb`	25.087	8.257	5.661
(5)	`wddl_0`	23.526	5.795	0.907
(6)	`wddl_1`	29.558	6.084	0.846
(7)	`wddl_2`	31.392	6.473	0.750
(8)	`wddl_4`	32.367	6.329	0.800
(9)	`ewddl_4`	40.250	8.050	1.054
(10)	`seclib_1`	14.824	4.556	0.766
(11)	`seclib_2`	13.978	4.889	0.837
(12)	`seclib_4`	11.897	4.404	0.729
(13)	`seclib_4ema`	15.593	4.681	0.806

– module (3), `stdcell_gb`, becomes drastically more secure.

It is remarkable that implementation (3) which is based on Guido Bertoni's architecture seems less vulnerable than the two other standard cell based implementations. This could be explained by the architecture. The architecture is in fact divided into three steps decode/permute/encode among which only the last encode step is input-dependent. It is based on a glitch-free 1-out-of-256 decomposition, that signs the same irrespective of the input, unless two consecutive inputs happen to be identical (in which rare case there is no dissipation at all). It demonstrates that a well-balanced architecture can reduce information leakage at a very low-cost in term of silicon area. The throughput is divided by two, which is anyway an overhead that dual-rail logics also have to pay for.

In the sequel, we study the metrics for only the 256 transitions corresponding to all possible 8-bit inputs preceded by a precharge phase to zero. As for dual-rail logic, Table E.5 also proves the importance of synchronization as SecLib seems more secure than WDDL (See Appendix E.7 for detailed power trace figures). It is however difficult to evaluate the gain of the differential routing on top of the differential placement. The only noting that holds for sure is that differential routing associated to shielding of dual pairs improves the security: (10) is indeed more dispersive than (12). This applies to SecLib, but not to WDDL, where the dispersion due to logic is the overwhelming source of dispersion: for WDDL, the more backend counter-measures, the larger the module, hence the more intense the information leakage.

One other remark is related to the metrics for `ewddl_4`. In fact, it was expected that the replacement of `AND` and `OR` gates by the `Enhanced-AND` and `Enhanced-OR` (Figure E.3) improves the symmetry of the design. But according to the measurements, this has increased the dispersion. This makes us tend to believe that early evaluation is predominant against technological asymmetry. Indeed, eWDDL, as WDDL, is prone to early evaluation; as eWDDL is based on more complex gates than WDDL (`MAJ` instead of `AND`/`OR`), the propagation time through the logic is increased [2], which exacerbates the early evaluation because it is cumulative along the combinatorial paths.

E.4.4.5 Confrontation With an Information Theoretic Metric

The level of robustness of a counter-measure can also be evaluated by the quantity of information it leaks. This approach requires an approximation of the probability distribution function (PDF) for one trace to actually match the correct input used during the acquisition. In our setup, we have a close to perfect estimation of the leakage trace for every possible input. By design, the computation of the substitution box is not disturbed by other unrelated activity and the high averaging rate of the oscilloscope greatly improves the signal's vertical resolution. However, it can be interesting to extrapolate the information available from each SubBytes block when the measurements are noisy, as in operational situations. The noise can, for instance, model the activity of surrounding logic gates, which will happen in practice, since SubBytes is customarily embedded into

2. In STM HCMOS9GPLL library, the average propagation time through the unload unitary `AND` (*resp.* `MAJ`) gate is 81 ps (*resp.* 146 ps).

a complete datapath with other substitution boxes. We thus introduce an artificial noise parameter σ. It is equal to the width of the PDFs, assumed to be Gaussians of identical variance σ^2 for any substitution box input.

Our evaluation is inspired from the one carried out by simulation on single logical gates [271] [3]. We replaced the simulations by the real measurements and the logic gates by a complete netlist of combinatorial gates making up the SubBytes instances. The dual-rail with precharge substitution boxes embedded in SubBytes correspond to the **Pre-Charged / not Masked Logic Styles** paragraph in Sec. 3.2 of [271]. Therefore, we compute the mutual information as per Eqn. (E.1), using notations of [271]:

$$\mathrm{I}(S_g, \mathbf{L}_{S_g}^{q=2^8}) = \mathrm{H}(S_g) - \mathrm{H}(S_g|\mathbf{L}_{S_g}^{q=2^8}) = \tag{E.1}$$
$$8 - \sum_{s_g=\texttt{0x00}}^{s_g=\texttt{0xff}} \Pr(s_g) \int_{\mathbf{l}} \Pr(\mathbf{l}|s_g) \log_2 \frac{\Pr(\mathbf{l}|s_g)}{\sum_s \Pr(\mathbf{l}|s)} \,\mathrm{d}\mathbf{l}\,.$$

We use for the input distribution $\Pr(s_g)$ a uniform law over $[\texttt{0x00}, \texttt{0xff}]$ and for $\Pr(\mathbf{l}|s_g)$ a multi-variate Gaussian distribution of mean the measurements and of covariance matrix a multiple of the identity of $]0, +\infty[^{T\times T}$.

The integration over all the samples is simplified by a principal component analysis (PCA) of the curves. Thanks to the pre-processing described in [13], we managed to replace all the initial samples of the curves by one single sample. The number of significative components in the PCA validates the limitation to one single sample; this makes it possible to simplify Eqn. (E.1) from a multi- to a single-valued integral.

The result is plotted in Fig. E.21. In this graph, the lowest curves are the most secure. It can be seen that the conclusions already drawn in Sec. E.4.4.4 still hold. The single-ended logics disclose more input bits than WDDL, that in turn is less secure than SecLib. We continue to note that the single-rail architecture of Guido Bertoni *et al.* performs almost as good as WDDL. Also, it appears clear the SecLib has a serious security improvement over WDDL. We also confirm that the eWDDL style does not improve WDDL, but instead makes it worse, certainly due to an exacerbated early evaluation propagation. Finally, some behavior amongst the SecLib modules are difficult to interpret, like for instance `seclib_2` that is less secure than the other SecLib modules, but for a narrow window of noise. It is nonetheless certain that SecLib with all the protections set (but without the M6-shield, namely `seclib_4`) is the most secure implementation. One final observation can be made: the $\mathrm{I}(S_g, \mathbf{L}_{S_g})$ curves for SecLib have a discontinuity when it is equal to 8 bits and the noise increases, whereas the behavior for WDDL, eWDDL and single-ended logics is continuous. This means that WDDL, eWDDL and single-ended logics have homogeneously distributed biases. At the opposite, SecLib traces have very few discrepancies when the inputs change: the discontinuity is probably due to a very small number of particularities for some rare inputs. This analysis shows that, should a designer be able to identify those discrepancies, the security level of SecLib could be easily improved.

3. This work has been extended recently on a four-bit datapath of PRESENT in [379].

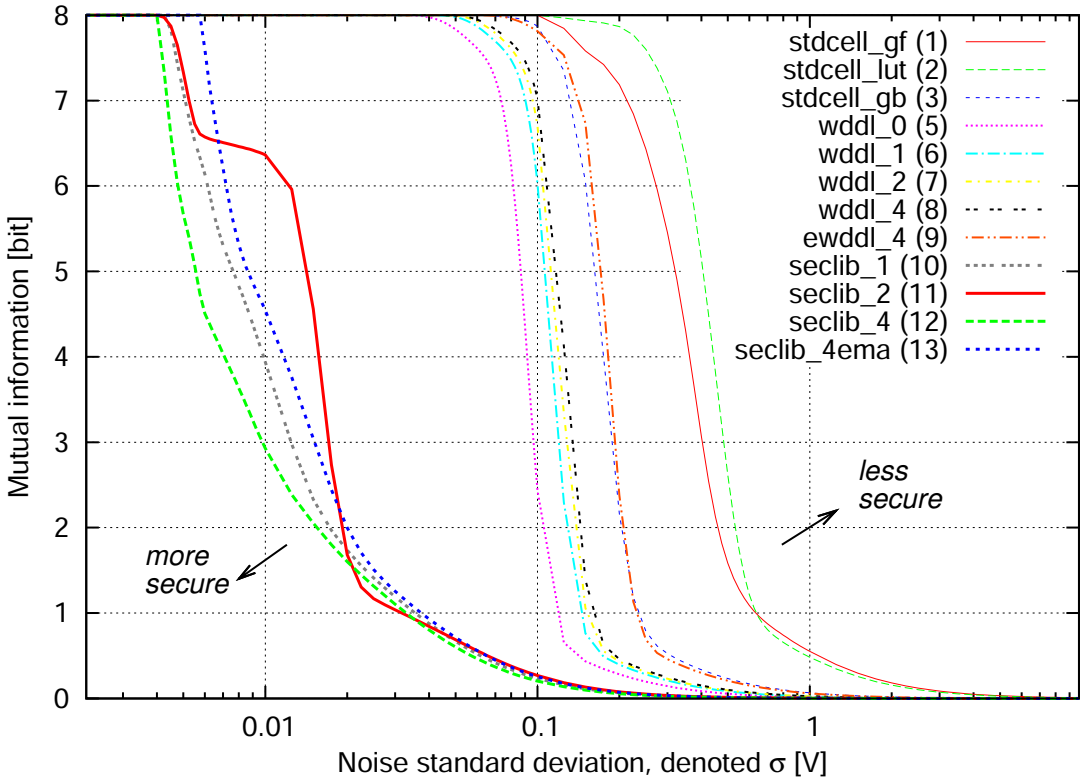


Figure E.21: Mutual information leaked by the implementations of SubBytes using the `0x00` spacer for precharge, in the hypothesis of noise homoscedasticity over all the different inputs.

E.5 Design-Time Security Evaluation and Backend-Level Counter-Measures Analysis

This section gathers the lessons learnt from the previous design-time (Sec. E.3) and *in silico* (Sec. E.4) evaluations. The efficiency of the logic styles and backend refinements is also discussed.

E.5.1 Reflections About High-Level Security Evaluation

High-level evaluations based on static analyses, such as [457] routing unbalancedness estimation, happen to be irrelevant. Indeed, experimental results show that for logics that do not synchronize the signals, the predominant source of unbalancedness is the relative arrival times of inputs. Depending on them and on the values of the inputs, the logic evaluates earlier or later. This early evaluation issue is thus a dynamic problem. It is several orders of magnitude more important than the dispersion of routing characteristics. A correct high-level security evaluation of non-synchronizing logics (such as AND-OR based logics) must thus resort to simulations or to techniques taking the timing behavior into account.

Notice that this remark does not apply to SecLib, since the very structure of this logic makes it possible to decouple the gates from their interconnection. Indeed, static (netlist-level) and dynamic (silicon-level) results agree.

The silicon-level measurements also revealed that amongst unprotected single-rail implementations of SubBytes, some can be almost as secure as WDDL or SecLib. The logic in question is that of Guido Bertoni: as every execution implies a decoding, all inputs activate roughly the same number of gates. Put differently, all execution paths are almost indiscernible: this appears clearly on Fig. E.1, where a typical execution path is highlighted. Whatever the input byte, the decoder sets only one bit amongst 256 to '1', that is driven to exactly 8/2 encoders (because SubBytes is balanced) all having the same structure. Therefore, even if this logic is larger than other unprotected descriptions, it remains smaller than WDDL and much smaller than SecLib circuits, for a comparable security level.

An other interesting point is about the M6-shielded SecLib instance. Eric Peeters already showed in the chapter 5 of his PhD thesis manuscript [352] that:

> "Metallic shield must be tamper resistant as well, because when connecting a differential probe on it, we were able to observe a data-dependent voltage. As a matter of fact, the metallic shield is turned into a very near-field electric probe."

We observe that a metallic shield increases the dissymmetry of an underneath DPL design. A "self-induction" effect might be the cause of such an effect. But for sure, the conclusion is that the usefulness of a top-level metallic shield is far from being obvious.

E.5.2 Summary About Security-Cost Trade-Offs

The previous analyses have made clear that some *would-be* counter-measures actually both increase the implementation cost and degrade the security level. This is case of eWDDL and the top-level electromagnetic shield. Those two solutions must positively be proscribed.

We note for the time that a non-protected single-rail logic can be made more security simply by interleaving every computation by a precharge to a constant value, such as `0x0`. The impact in terms of silicon area is negligible, but the throughput is divided by two. The other counter-measures, labeled $\mathcal{B}1$ to $\mathcal{B}5$, increase the security level. However, they are actually useful only if the logic is immune to early evaluation. SecLib is in the *silicon-domain* (as opposed to the *wire-domain*), which means that the area of the cells is limiting the density and not the congestions in the interconnect resources. Therefore, in the case of SecLib, The gain they convey by the accumulation of security features is visible in terms of security, and in the meantime also free in hardware, since $\mathcal{B}3$ to $\mathcal{B}5$ complexify the routing, which is not a critical resource.

E.5.3 Suitability of an Elementary Pattern Circuits for Security Evaluations

The backend-level improvements do not translate into an observable security increase as for WDDL, because we identified that the early evaluation is overwhelmingly the predominant dispersive feature. Nonetheless, we could have expected SecLib to disclose improvements with the backend design care. Paradoxically enough, it is not straightforward to appreciate the impact of backend features on SecLib dispersion. This might be due to the over-simplification of the design; if the SubBytes instances were not insulated (not from the substrate noise but from other noisy instances by a large on-chip spacing), they would be more coupled with extrinsic activity (referred to as "algorithmic noise" in the context of attacks against cryptoprocessors [60, 217]). In this case, we could observe that SubBytes instances with poor backend features would be more influenced by this coupling than full-featured SecLib SubBytes instances. Unfortunately, we cannot verify this hypothesis on the ASIC: do poorly routed and unshielded SecLib instances appear more secure than they really are because of an evaluation artifact?

E.6 Conclusions and Perspectives

E.6.1 Conclusions

DPL styles are designed and used to counter-act DPA attacks by making the power consumption constant. There are several DPL logics such as WDDL and SecLib, respectively based on standard cells and totally customized cells forcing signals synchronization. In this paper we compare these two logics by analyzing the power dispersion of a combinatorial block, the AES substitution box (SubBytes). Our analysis demonstrates that dual-rail logic implementations are indisputably more secured than single-rail logics. We

find out that choosing a balanced architecture such as described by Guido Bertoni *et al.* combined with a precharge to zero does reduce the power dispersion impressively, thus increasing the security level against power analysis attacks. We also demonstrate that SecLib is less dispersive than WDDL, confirming experimentally that signals synchronization is important to avoid data-dependent early evaluation and precharge. The security benefits of second-order countermeasures, such as differential placement, routing, dummies and shield against cross-talk are observed on SecLib.

E.6.2 Perspectives

As static high-level security evaluations are not accurate enough, netlist temporal simulation must be used instead for pre-fabrication validation purposes. This approach has been initiated for instance in [174] with logic simulation (ideal transitions). To further model signals slopes, fast gate-level or transistor-level simulations are mandatory. Efforts in this direction have already been deployed, *e.g.* by Huiyun Li *et al.* [262] or by Giorgio Di Natale *et al.* [335].

We emit the hypothesis that results on SecLib instances of `SubBytes` were evaluated optimistically because of the absence of neighbour logic, and that the impact of coupling cannot be assessed. We suggest to consider FPGAs as prototyping platforms: FPGAs do not exactly behave SCA-wise as ASICs (even at constant technology); nevertheless they allow to better iterate and test more configurations. For instance, the SASEBO boards [391] with the EveSoC environment [224] can be such a commodity.

E.7 Appendix: Traces Showing Power Dispersion for Twelve Implementations of `SubBytes`

Figures E.22 and E.23 show the power dispersion measured for the 256 possible inputs ($\forall f \in [\texttt{0x00}, \texttt{0xff}], \texttt{0x00} \rightarrow f$) respectively for standard cell logic, and dual-rail logic — WDDL *versus* SecLib.

The acquisition chain characteristics are listed below:

- The probe's bandwidth is 5 GHz;
- The sampling rate of the acquisition apparatus (Infiniium 54 855A sold by Agilent) is 20 Gsample/s;
- The vertical caliber is 1 mV;
- The curves are averaged 256 times by the oscilloscope, leading to 12-bit vertical resolution;

The traces are displayed raw: no post-processing has been done to correct their shape. Compared to a crypto-processor's regular trace (such as the example given in Fig. E.6), the average is non-zero after evaluation. This is due to the fact that the SubBytes modules, the power consumption of which is measured, are not electrically insulated from the rest of the `SubBytes` internal logic. Hence a cross-coupling between several parts of the silicon die, that induce a background noise. As the same programmation sequence is employed to test every SubBytes block, the cross-coupling effect is a constant

phenomenon that merely adds up to the relevant measurements. Because it is the same irrespectively of the addressed SubBytes module, this "continuous component" can safely be ignored.

Acknowledgments

This work has been partly financed by the french conseil régional "Provence Alpes Côte d'Azur" (Région PACA) and by the SCS (Solutions Communicantes Sécurisées) competitivity cluster via the CALISSON project. We are grateful to STMicroelectronics AST (Advanced System Technology) department for having launched and encouraged this project, to CNFM (Coordination Nationale pour la Formation en Micro et nanoélectronique) for CAD tools licenses and to CMP (Circuits Multi-Projets) for subcontracting the chip fabrication and packaging. We thank Karim Benkalaia, from COMELEC department of TELECOM ParisTech, for the design and the test PCB for deported ICs, such as `SubBytes`. We acknowledge interesting discussions with Guido Bertoni, Jean-Luc Danger and Yves Mathieu, as well as relevent suggestions of improvements from the anonymous reviewers.

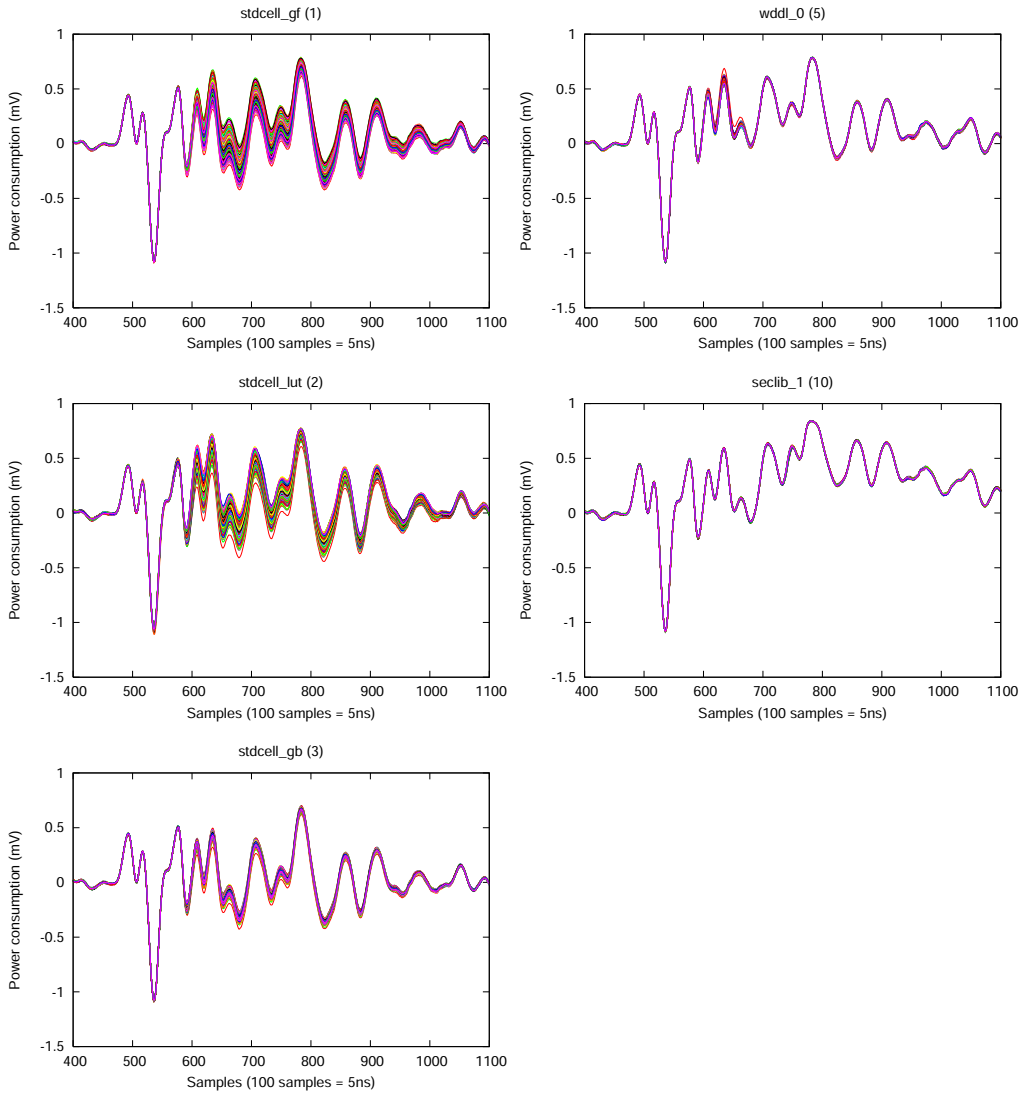


Figure E.22: Power traces for 256 inputs with `0x0` or `0x00` precharge — comparison between standard cell logics and dual-rail logics.

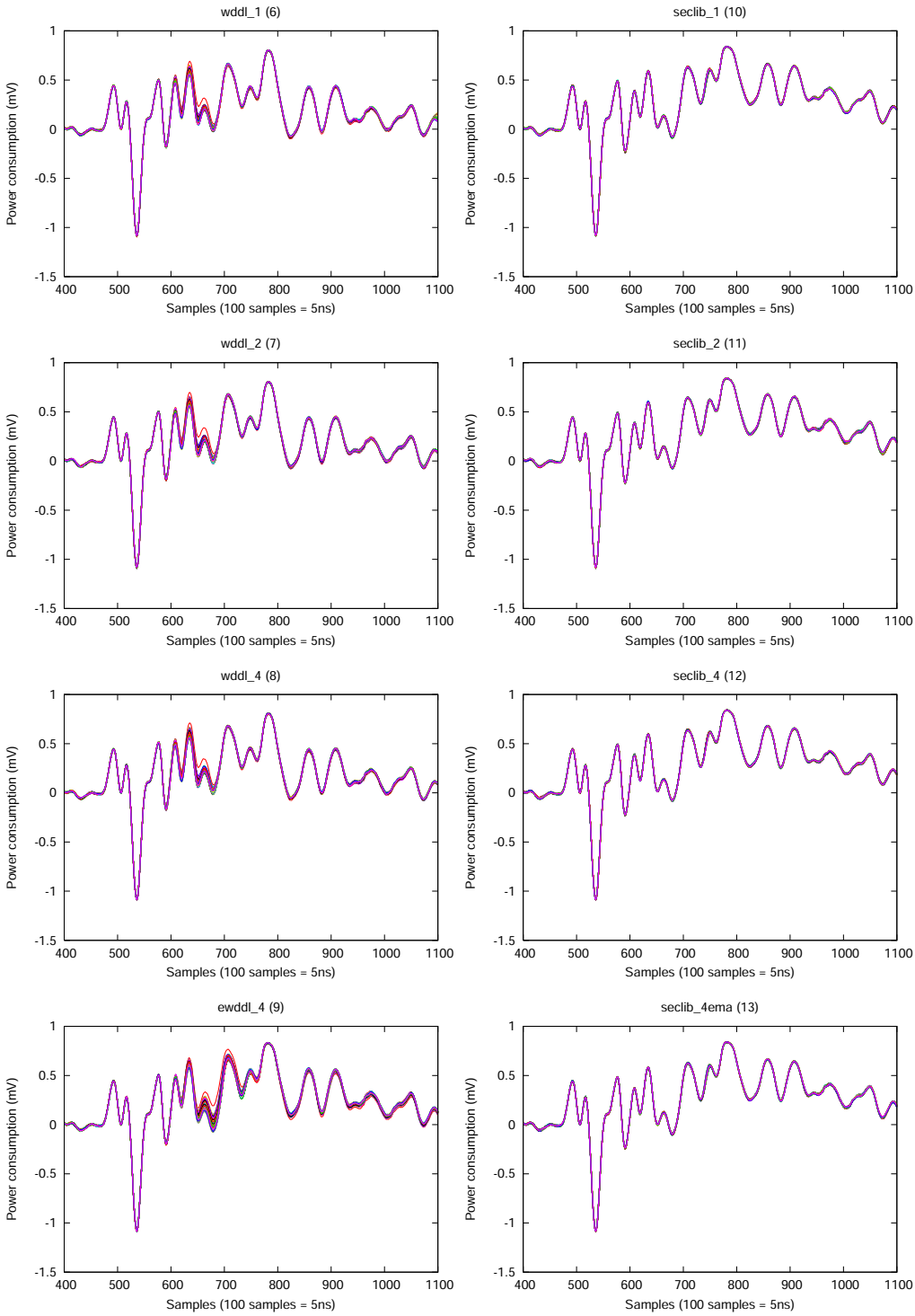


Figure E.23: Power traces for 256 inputs with `0x00` precharge — comparison between WDDL and SecLib.

Appendix F

Unrolling Cryptographic Circuits: A Simple Countermeasure Against Side-Channel Attacks

Extended version of article [39]

Authors: *Shivam Bhasin, Sylvain Guilley, Laurent Sauvage and Jean-Luc Danger*

Abstract

Cryptographic cores are used to protect various devices but their physical implementation can be compromised by observing dynamic circuit emanations in order to derive information about the secrets it conceals. Protection against these attacks, also called side channel attacks are major concern of the cryptographic community. Masking and dual-rail precharge logic are promoted as its countermeasures but each has its own vulnerabilities. In this article, we propose a simple countermeasure which comprises unrolling rounds of a cryptographic algorithm such that multiple rounds are executed per clock cycle. This will require a stronger hypothesis on multiple bits due to deeper diffusion of the key. Results show that it resist against correlation power analysis on Hamming distance and Hamming weight model if the datapath is cleared after each operation. We also evaluated mutual information metric on the design and results show that unrolled DES is less vulnerable.

Keywords: Data encryption standard, side-channel attack, architectural countermeasure, mutual information metric.

F.1 Introduction

With the generalization of open networks, information society regards security as a critical factor. Modern cryptographic algorithms which ensure security are robust and

free from practical cryptanalysis. However, other methods which target the physical implementation of an algorithm can be deployed to break the security. These attacks can be mounted by merely observing or perturbing the targeted system. Observing the activity of the system and its correlation with potential guesses can yield sensible information. Such attacks are better known as Side Channel Attacks (SCAs) [249]. When a device is perturbed such that it yields a non-nominal output, this together with expected output can lead to the secret key. Such attacks are called as Differential Fault Analyses (DFAs) [45]. The passive attacks that consist in observing the chip are difficult to protect since the chip is even not aware of the attack. Therefore these attacks are considered more critical.

SCAs try to recognize synchronous operations (rounds of cryptographic operations) in the leakage of a device. Then for a chosen round, the leakage is correlated with some guesses to reveal secret information. It is possible to guess some key bits because the value of key remains same for one or a set of synchronous operations. For example if we consider DES, cryptanalysis is impractical as we need a huge number of plaintext or ciphertext. Whereas with power attacks only the power consumption of a few hundreds of encryption are needed to break a non-protected implementation. For instance in DPA contest [445], the participants have demonstrated that DES could be broken in 141 traces in average. Therefore it is essential to protect implementations against SCA.

State of the art countermeasures can be widely classified into two categories *i.e.* information making and information hiding. Masking [6] countermeasures rely on confusing the attacker. A random generated mask is used while running the algorithm such as the mask affects the intermediate states without affecting the end result. Owing to this technique, the attacker observes leakage corresponding to mask and not the actual key bits. Although a nicely masked circuit can resist first order SCA but higher order SCA can still compromise the security of the design

Information hiding as the name suggests hides the information from attacker. The algorithm is implemented in such a way that leakage remains constant irrespective of the computations performed. Dual-rail precharge logic (DPL) [456] is a countermeasure based on information hiding. The principle of this countermeasure is to generate a design equivalent and with opposite behaviour of the target design such that every part of the circuit is perfectly balanced. This way the activity of the doubled design remains constant. There are some countermeasures which combine hiding and masking techniques in order to achieve higher level of security. The major problem of these countermeasures is that it is hard to design a perfectly balanced circuit. Even minor imbalance in space (unbalanced dual nets) or time (early evaluation) can be exploited by sophisticated attacking techniques to reveal sensitive information.

In [435], the effect of pipelining on security is studied. In this article, we investigate the other trend, namely pipelining less; this way, all registers become unpredictable depending on the key (*i.e.* a hypothesis test involves too many key hypotheses). The idea is to implement the design in such a way that the key changes more than once during a synchronous operation. In other words, more than one round of a cryptographic algorithm are executed in one synchronous operation. The rest of the paper is organized

as follows. Section F.2 explains the theory of the proposed countermeasure. It also details the implementation details of a fully unrolled DES. Section F.3 evaluates fully unrolled DES against the iterative DES using correlation power analysis (CPA [60]). Finally, section F.4 concludes the paper.

F.2 Proposed Countermeasure

F.2.1 Rationale of the Countermeasure

In a cryptographic block product algorithm, data is ciphered by repeating a set of operations with a different key value each time generated from the previous key. These set of operations are called as rounds. The number of rounds are chosen such that linear and differential cryptanalysis are more difficult than an exhaustive key search. Normally, cryptographic circuits are designed to perform either some operations of a round or the whole round in one clock cycle. Thus the value of the key remains the same for one or more clock cycles. The attacker can guess some of the key bits and correlate it with leakage acquired. A correct guess will give a much higher correlation as compared to wrong guesses.

Most of the traditional SCA attacks target the registers where the result of each round is stored. This is because the leakage from the register is high due to its load and the leakage is synchronised to the clock. In combinatorial logic, the leakage is low and spread over time. If the result of a round is stored in the register at the end of each clock cycle, attacker can easily retrieve the subkey by guessing and correlating. Now, if the key is changed more than once during one clock cycle *i.e.* multiple rounds are executed per clock cycle the key used for one round is further diffused deeper into the design and mixed with the second key and so on. Thus exploiting this property we propose to design the cryptographic coprocessors in such a way that it executes multiple rounds in one clock cycle. We call this as unrolling the rounds of the algorithm. Also we define unrolling factor as the number of rounds unrolled. An implementation unrolled twice means that two rounds are performed at every clock cycle. A didactic presentation of the loop unrolling technique is given by Kris Gaj and Pawel Chodowiec in the chapter 10 of [244], along with a discussion about its pros and cons from a performance point of view.

Figure F.1(a) shows the architecture of one round of a normal iterative cryptographic algorithm while figure F.1(b) shows the architecture of an unrolled cryptographic algorithm. An idea of the difficulty to mount a side channel attack on the unrolled version can be estimated from the following discussion. Suppose, we have two implementations of a cryptographic algorithm: one iterative and the other unrolled with an unrolling factor of 2 as shown in fig F.1(a) and (b) respectively. Let us see the signal and the noise when the attack is mounted on 1-bit. In the iterative design, the signal will be the sum of the power activity of all the combinatorial gates and flip-flop involved in calculating that bit. The noise shall be sum of power activity of other gates and flip-flops. In the unrolled design, if we implement an attack on 1-bit in the first of the two rounds, the

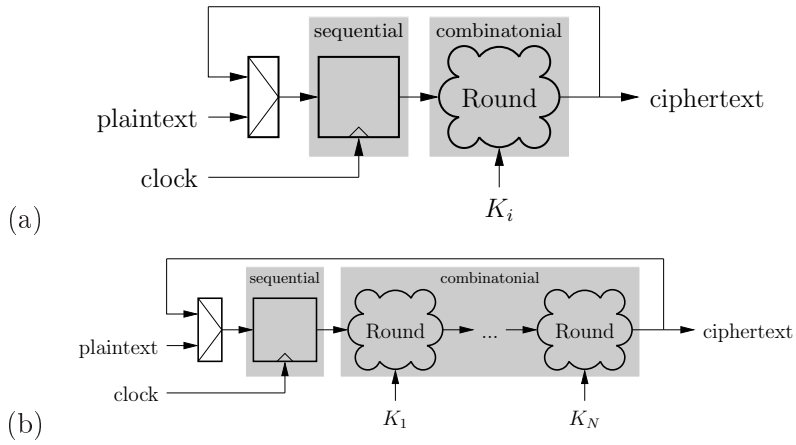


Figure F.1: (a) Architecture of a iterative cryptographic algorithm. (b) Architecture of a fully unrolled cryptographic algorithm.

signal will be the power activity of the gates involved only as the result is not memorised. The noise shall be twice the previous value as components are doubled. As explained before the power activity of a combinatorial gates is lesser than the power activity of a register. This results in SNR reduction of more than twice.

A rough evaluation of the theoretical complexity of this countermeasure in terms of area is given by the unrolling factor. Thus a design unrolled twice will have double the area of its original design as far as combinatorial part is concerned. In terms of performance, the trade-off is almost the same as original design. Unrolling factor of n will multiply the critical path by n times and thus maximum frequency is reduced $1/n$ times. Since n rounds are executed per clock cycle, N/n clock cycles are needed to execute the whole algorithm where N is the total number of rounds. Thus the throughput is approximately the same for original and unrolled design. The practical results are better than the one described below as some of the unnecessary components like multiplexers are removed while unrolling. Thus the area is less than n times and the operating frequency is more than $1/n$ times. We also point out that the unrolling does not impact the possibility of the encrypting block to be used in any mode of operation (CBC, CFB, OFB, *etc.*).

Fully unrolled DES implementation: An iterative architecture can be made combinatorial, by removing its register transfers occurring during the rounds [171]. In the case of DES, the algorithm combinatorial depth is thus roughly increased by a factor of sixteen, but the registers LR and CD remain frozen during sixteen clock cycles, which makes up for the delay through the gates. The architecture, based on that described in [195], and the floorplan are depicted in Fig. F.2(a) and (b). It is a special case of the so called *brutal countermeasure* mentioned in [387], where the "glued blocks" actually make up the entire datapath. The inputs 1 of the LR multiplexer and 2 of the CD multiplexer play the role of enable for the corresponding registers. The key schedule

(a) (b)

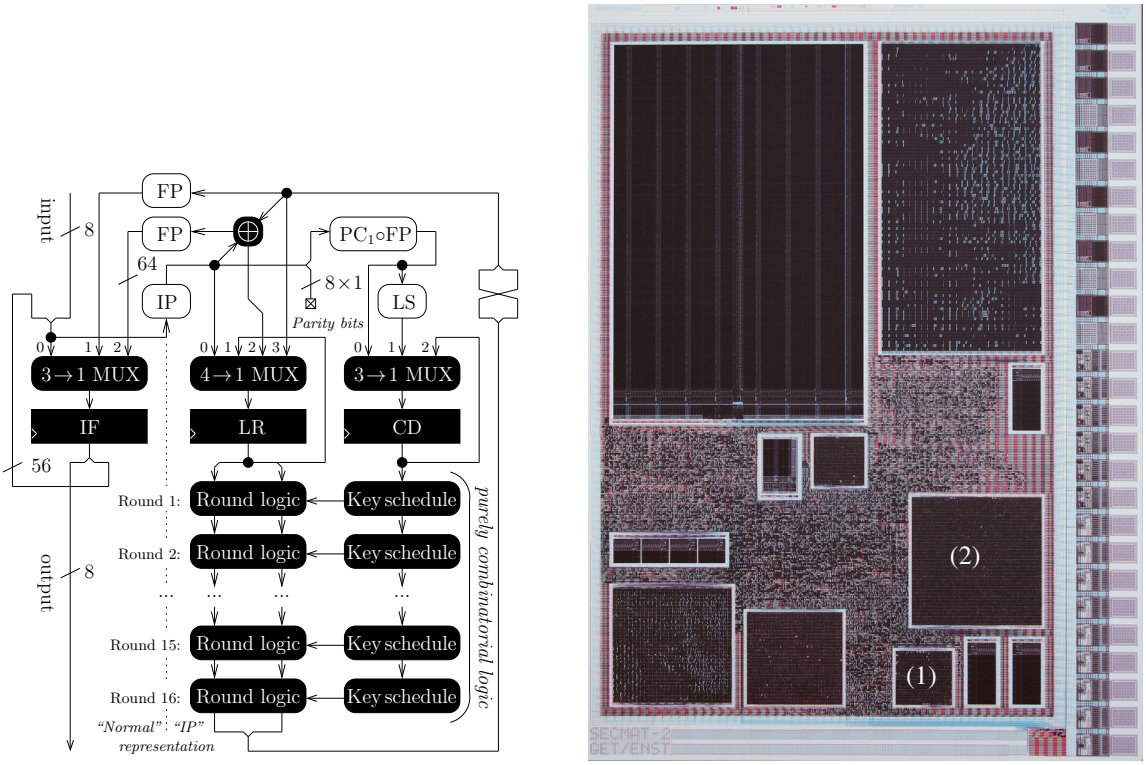


Figure F.2: (a) Unrolled DES Architecture. (b) Floorplan of the ASIC implementing DES iterative **(1)** and DES unrolled **(2)**.

consists in a sequence of pre-computed circular shifts which can be implemented just by switching wires and requires no logic. Such a technique is only valid for certain algorithms like DES and the absence of logic in key schedule avoids leakage. Thus attacks like [2] cannot be mounted anymore.

The synthesizers, in default mode, attempt to fit a timing path into one clock cycle. To synthesize such a design there is need to relax the timing constraints. In the combinatorial DES specific case, the logic driven by LR and CD has time equivalent to sixteen clock cycles to execute. This piece of information cannot be easily inferred, thus user constraints must be set. They basically consist in specifying spare clock cycles for some timing arcs. The timing paths that are concerned thus start at registers LR and CD, plus the Boolean signal originating from the control that tells whether the current operation is a ciphering or a deciphering , where the shifts can be interpreted left or right-wise. The "multi-cycle" constraints listed in Fig. F.3 express the fact that outputs of LR and CD are sixteen times slower that the clock and that the signal to decide between ciphering and deciphering is a false timing path. This last path is indeed never critical because the choice between encryption and decryption is not modified during one computation.

```
set_current_module des_datapath_combi_wrapper; # Internal constraints
set_current_instance [find -hier -inst I_REG_LR];
# The following constraint (1+15 cycles allowed for the computation)
# concerns the whole bus:
set_cycle_addition -from [get_info [lindex [find -port q] 0] bus] 15;
set_current_instance [find -hier -inst I_REG_CD];
set_cycle_addition -from [get_info [lindex [find -port q] 0] bus] 15;
set_current_module des_datapath_combi; # External constraint
set_false_path -from [find -port sel_left_not_right]; # Encrypt/Decrypt
```

Figure F.3: TCL timing constraints crafted for the "multi-cycle" DES combinatorial datapath synthesis by Cadence `bgx_shell`.

The key schedule can be implemented by mere routing of wires, with no logic usage. Indeed, every round key in DES is obtained by simply selecting the adequate bits from the 56 bit master key. However, this peculiar property applies to DES only and cannot be generalized for all the cryptographic algorithms.

F.3 Experimental Results

We implemented an iterative DES and a fully unrolled DES on SecMatV2: an academic ASIC for security evaluation of cryptoprocessors implemented in 130 nm technology from STMicroelectronics. The placement constraint used for both modules is that their placement density is 95%. Therefore we found that iterative DES consumes an area of 24787 μm^2 while the unrolled DES consumes an area of 139816 μm^2. The ratio in terms of surface is thus as low as 5.64 lower than expected *i.e.* 16, the unrolling factor which is due to removal of registers, removal of logic involved in the iteration management (multiplexers), round boundaries optimization. Also the key schedule is completely dissolved in mere routing which is a property specific to DES algorithm. In terms of performance for a nominal operating frequency, the iterative DES needs almost 5 times more time for single encryption. However, the operating frequency is not the maximal operating frequency in this case.

The average side-channel curves for one DES encryption are shown in Fig. F.4(a) and F.4(b) respectively for the iterative reference DES and the combinatorial instance. It clearly appears in Fig. F.4 that the variations increase during the encryption.

Side-channel attacks can be roughly divided into two categories. On one hand correlation attacks make the assumption of a known leakage model; several models corresponding to different values of the secret are devised. The model that correlate the better with the concrete measurements discloses the secret. On the other hand, template attacks divide into two steps. The first step is done off-line; it consists in pre-characterizing the circuit in an almost blind fashion, for as many representative values of the message and key inputs. Stochastic attacks are a variant where the pre-characterization is made more simple by injecting some partial knowledge about the target's leakage. The second step is the on-line attack proper. The attacker attempts to recognize the secret by matching

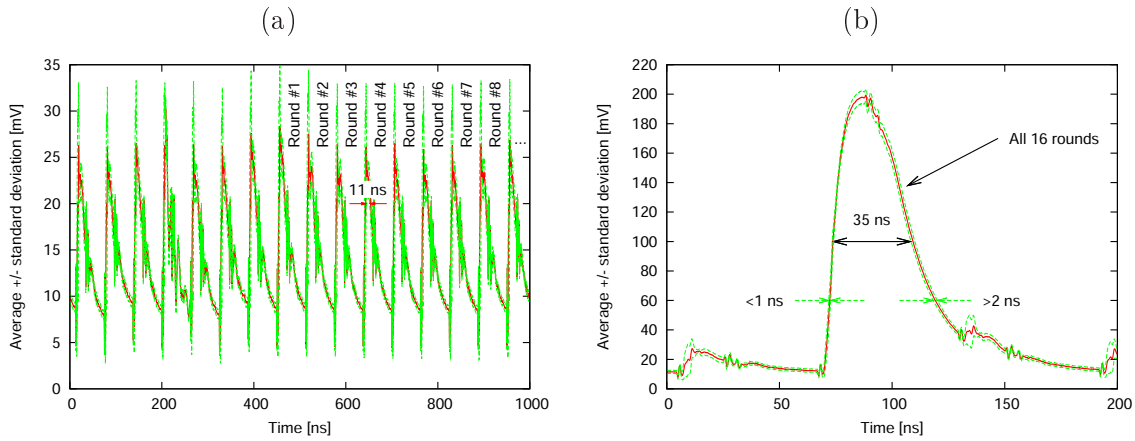


Figure F.4: (a) Sequential iterative DES encryption signature, with the average variation margin, for statistics collected on 10k measurements. (b) Average combinatorial DES encryption signature, with the average variation margin, for statistics collected on 100k measurements.

measurements obtained from a fixed albeit unknown secret key.

We show that correlation attacks are made very implausible on a fully combinatorial implementation, due to the signal's desynchronization, even in the early rounds (represented in Fig. F.5). First of all, we apply the same attack that is successful on the iterative reference implementation. It consists in a correlation of the measurements with the consecutive values of the right datapath register R_0, that leaks $\mathcal{L}(initial : R_0, final : L_0 \oplus f(R_0, K_1)) = |R_0 \oplus L_0 \oplus f(R_0, K_1)|$. The attack results on DES iterative and unrolled are shown in Tab. F.1 and F.2 respectively . Without any surprise, this attack completely fails on the combinatorial instance of DES, since the targeted transition has disappeared in the unrolled implementation. We would like to emphasize that each time a encryption is done, the datapath should be cleared. This can be done like precharge in DPL or by propagating random values without interference from the key. This is because, if two consecutive computations are done then some correlation can be found on the basis of previous computation.

F.3.1 Attack on the Unrolled DES

Now let us see a case when the previously described constraints are not respected i.e. two encryption are done without clearing the datapath. We explore two leakage models, namely the Hamming weight (HW) and the Hamming distance (HD), on two neuralgic positions of the algorithm, namely the Feistel function output (P1) and the round output right half (P2). We find that the HD on P1 completely discloses the key. The results are given in Tab. F.3. We can see that for all the eight broken substitution boxes, the signal-to-noise ratio (SNR) is much smaller than for the case of the reference circuit. The results for the sbox 4 are printed in italics, because actually two keys are guessed simultaneously in a unrolled implementation, due to a mathematical property of this

Table F.1: Key recovery attack on the iterative reference DES using a CPA over 10K traces.

Sbox index	Key		Lock_t $0 \leq \cdot \leq 10\,000$	SNR	Max CPA [%]
	Actual	Guessed			
1	56	56	4 314	4.38603	8.40
2	11	11	7 848	3.94818	5.68
3	59	59	1 247	5.29027	6.81
4	38	38	3 555	5.09747	5.94
5	0	0	2 272	7.25941	8.86
6	13	13	3 868	4.52662	8.10
7	25	25	4 399	4.69634	6.28
8	55	55	273	6.81590	14.68

Table F.2: Key recovery attack on the unrolled DES using a CPA over 100K traces.

Sbox index	Key		Lock_t $0 \leq \cdot \leq 100\,000$	SNR	Max CPA [%]
	Actual	Guessed			
1	56	58	87 976	1.83827	3.25
2	11	21	75 073	3.04394	1.52
3	59	17	97 462	2.07826	2.69
4	38	25	71 369	1.63005	4.85
5	0	53	70 590	3.45533	2.18
6	13	26	99 982	3.01725	1.18
7	25	22	70 433	2.07131	3.37
8	55	47	74 552	2.78395	3.26

sbox detailed in appendix F.5. The fourth sbox S_4 of DES has the following property: $\forall x, y \in \{0,1\}^6$, $S_4(x) \oplus S_4(y)$ and $S_4(x \oplus \texttt{0x2f}) \oplus S_4(y \oplus \texttt{0x2f})$ are palindromic. This fact can be shown by computing exhaustively the two expressions and comparing them.

Therefore, we have a remarkable Hamming distance conservation property: $\forall x, y \in \{0,1\}^6$, $|S_4(x) \oplus S_4(y)| = |S_4(x \oplus \texttt{0x2f}) \oplus S_4(y \oplus \texttt{0x2f})|$. As a conclusion, in a Hamming distance model, two keys are retrieved in pairs: the correct one and one another (false), equal to the correct key translated by `0x2f`.

To show that the correlations of the sboxes output (locus P1) are very disrupted due to their combinatorial nature, we have computed the DPA peaks, shown in Fig. F.6. We favor DPA [248] over CPA [60], because, as explained in the technical article [196], the covariance used by DPA extracts the activity of some nets in the netlist, which is interesting for leakage characterization. As for the CPA, it is more suitable for attacks,

Table F.3: Key recovery attack using the a CPA with a Hamming distance model (with respect to the previous encryption) over 100K traces.

Sbox	**Key**		**Lock_t**	**SNR**	**Max CPA**
index	**Actual**	**Guessed**	$0 \leq \cdot \leq 100\,000$		[%]
1	56	56	16 557	2.20267	2.17
2	11	11	44 092	2.15008	2.09
3	59	59	36 090	2.50697	2.22
4	*38*	*9*	*3 291*	*3.73242*	*5.01*
5	0	0	27 164	1.96649	2.28
6	13	13	20 138	2.13591	2.65
7	25	25	17 862	2.11245	2.86
8	55	55	37 317	2.77701	2.75

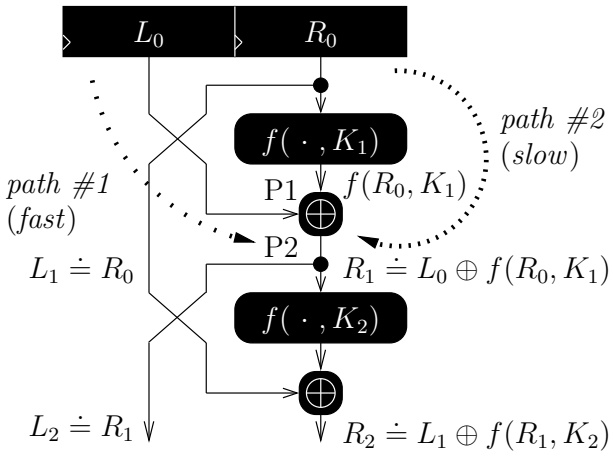


Figure F.5: Notations used to describe the combinatorial DES leakage functions.

because the normalization by the trace standard deviation corrects the fact that the leakage is not necessarily maximum at the times where the side-channel is [208]. The DPA covariance $|f(R_r^{-1}, K_{r+1}) \oplus f(R_r, K_{r+1})|$ for all $r \in [0, 6]$ are plotted in Fig. F.6. We have also added the transition in R_0 between two consecutive messages, because it indicates the computation beginning and its end. The beginning consists of the R_0 register sampling at the rising edge of the clock. The end corresponds to the other transition (final $\rightarrow$ initial), in the R_0 register input latches, that are transparent, and that dissipate even in the absence of a clock event. We observe that the DPA covariances do not especially show peaks ordered in time. This indicates the link between the data and the side-channel measurement is destroyed as early as the first couple of rounds.

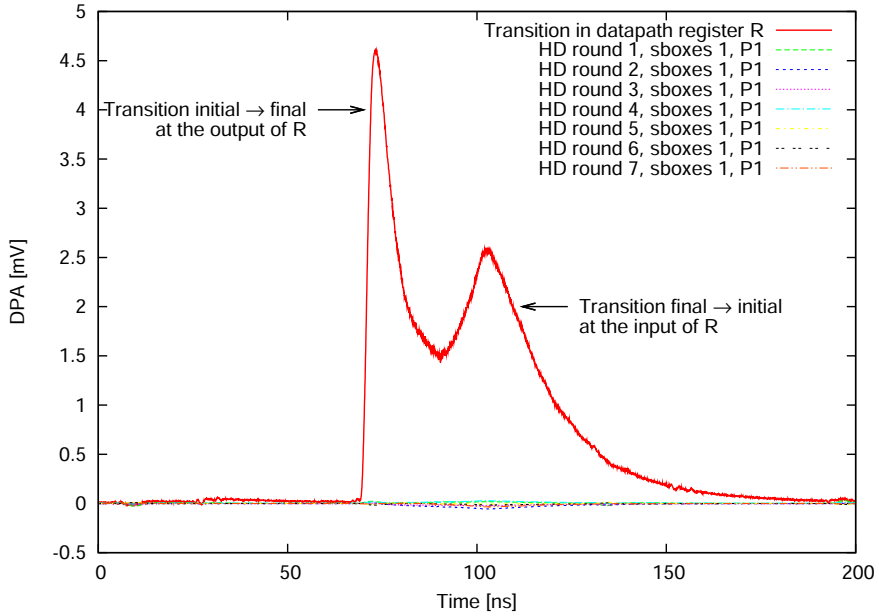


Figure F.6: DPA covariance for the register transfer R_0, and round correlations for the first sbox outputs.

To conclude with the security analysis, we discuss briefly on the unsuitability of other SCAs. Template attacks are expected to become less a concern as technology typical feature sizes shrink and characteristics dispersion increases [370]. Preliminary works on 130 nm technologies [189] suggest that the intra-die technological mismatches are the preponderant source of variation, surpassing the imperfections of the logic style.

F.3.2 Evaluation Based on Mutual Information Metric

Mutual information analysis (MIA) has been introduced in [141] and further discussed in [364]. This analysis captures whatsoever dependence between measurements and a leakage model. It is thus a tool suited for an information leakage evaluation, as pointed out in [468]. The default leakage model does not assume any device-specific knowledge. Therefore it considers plain dependency with one sensitive and predicable word within the device. The notions of sensitivity and predictability have been defined in [437]. Basically, a variable is sensitive if it depends on one secret, and predictable if testing all the hypotheses for this variable is computationally tractable. The leakage-agnostic approach is the one employed in template attacks [69].

We have computed the mutual information (MI) between the right half of the datapath for sbox #1 and each point of our experimental traces. The results are plotted in Fig. F.7 for the 80k traces of the iterative DES module and the 100k traces of the unrolled one. In the iterative circuit, the MI is roughly the same for each round. How-

ever, it depends on the round index for the combinatorial circuit; therefore we represent a couple of them in Fig. F.7. It appears clearly that the sequential circuit is leaking more information about the first round than the combinatorial. Hence the vulnerability is less significant for our proposed countermeasure.

F.4 Conclusion and Perspectives

Information masking and hiding are two protection techniques against side-channel attacks. We propose a new countermeasure which comprises unrolling of rounds of a cryptographic algorithm to execute during a single clock. Results show that unrolling is secure against power attacks with a constraint of clearing the datapath after each encryption. We also evaluated mutual information metric on the design and results show that unrolled DES is less vulnerable. Further work involves testing this countermeasure with other algorithms like AES, *etc.* Also it could be interesting to partially unroll the algorithm like the rounds which are soft targets for an attacker.

Finally, we mention the potential advantage of algorithms unrolling against some fault attacks; for instance, it is impossible to inject faults via a setup time violation [122, 412, 242], produced by either under-powering or over-clocking the unrolled module. The resistance of partially or completely unrolled architectures against other DFAs is thus an interesting research direction.

Acknowledgments

This work has been partly financed by the french national research agency (ANR), through the `ANR-07-ARFU-010` grant "SeFPGA" (Secured Embedded FPGAs). We acknowledge interesting discussions and encouragements with Renaud Pacalet from the LabSoC laboratory of TELECOM ParisTech at Sophia-Antipolis.

F.5 Appendix: Equiprobable Keys For DES Sbox 4 in an Unrolled Implementation

Let us denote $\mathbb{1}_i$ the Boolean vector of $\{0,1\}^4$ that is equal to zero everywhere but at position $i \in [\![0,3]\!]$. The Boolean application $\{0,1\}^4 \to \{0,1\}, v \mapsto \mathbb{1}_i \cdot v$ is the selection of coordinate i. The fourth sbox S_4 of DES enjoys the following remarkable property:

$$\begin{aligned} S_4XP &\doteq \left(\sum_{x\in\{0,1\}^6} (-1)^{\mathbb{1}_i\cdot S_4(x)\oplus\mathbb{1}_j\cdot S_4(x\oplus \texttt{0x2f})} \right)_{0\le i,j\le 3} \\ &= 2^6 \times \begin{pmatrix} 0 & 0 & 0 & +1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ +1 & 0 & 0 & 0 \end{pmatrix}_{0\le i,j\le 3} . \end{aligned} \tag{F.1}$$

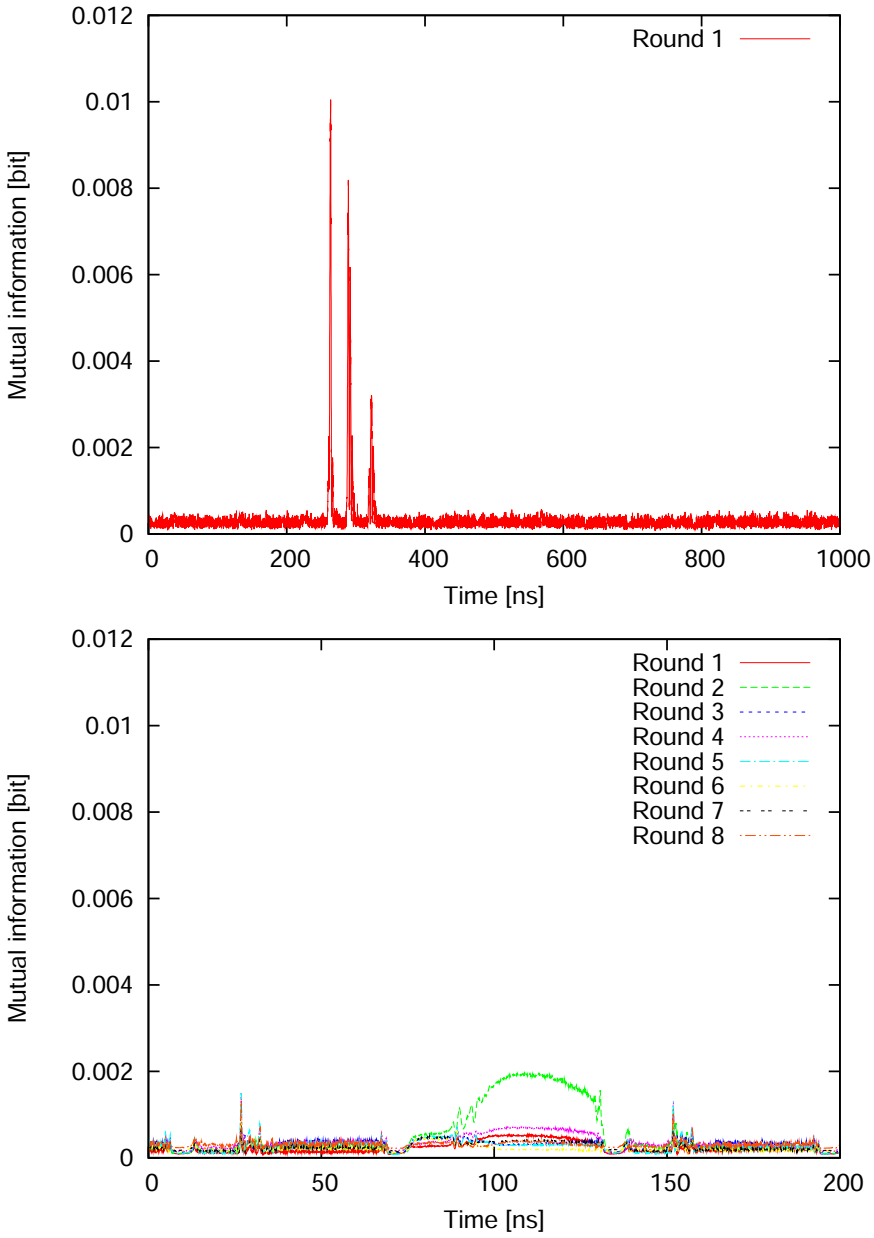


Figure F.7: Mutual information metric for sequential (*top*) and combinatorial (*bottom*) DES.

This fact can be shown by computing exhaustively the two expressions and comparing them. In fact, it is sufficient to compute $\mathrm{S_4XP}_{0,3}$ and $\mathrm{S_4XP}_{1,2}$. Indeed, the matrix is symmetric ($\mathrm{S_4XP}_{i,j} = \mathrm{S_4XP}_{j,i}$), and knowing that 4 coefficients are equal to $\pm 2^6$, the others are equal to zero. The reason is that the norm-2 of the matrix $||\mathrm{S_4XP}||_2 \doteq \sum_{i,j} \mathrm{S_4XP}_{i,j}^2$ is smaller or equal to $4 \times 2^{2\times 6}$.

Here is the proof: let f be a balanced (n,m)-function, and let $\mathsf{Offset} \in \{0,1\}^6$ be an input:

$$\sum_{\substack{0\le i<m\\ 0\le j<m}} \left(\sum_{x\in\{0,1\}^n} (-1)^{\mathbb{1}_i\cdot f(x)\oplus \mathbb{1}_j\cdot f(x\oplus \mathsf{Offset})} \right)^2 \quad \left[\begin{array}{c}\text{This is } ||\mathrm{S_4XP}||_2\\ \text{when } \mathsf{Offset} = \mathtt{0x2f}\end{array}\right] \tag{F.2}$$

$$= \sum_{x,y}\sum_{i,j} (-1)^{\mathbb{1}_i\cdot f(x)\oplus \mathbb{1}_j\cdot f(x\oplus \mathsf{Offset})\oplus \mathbb{1}_i\cdot f(y)\oplus \mathbb{1}_j\cdot f(y\oplus \mathsf{Offset})}$$

$$= \sum_{x,y} \left(\sum_i (-1)^{\mathbb{1}_i\cdot f(x)\oplus \mathbb{1}_j\cdot f(y)} \right) \cdot \left(\sum_j (-1)^{\mathbb{1}_j\cdot f(y\oplus \mathsf{Offset})\oplus \mathbb{1}_j\cdot f(y\oplus \mathsf{Offset})} \right)$$

$$= \sum_{z\in\{0,1\}^{2n}} \phi(z)\cdot\phi(z\oplus \mathsf{Offset}) \quad \text{where: } \begin{array}{c}\phi : \{0,1\}^{2n} \to \mathbb{Z}\\ z=(x,y) \mapsto \sum_i (-1)^{\mathbb{1}_i\cdot(f(x)\oplus f(y))}\end{array}$$

$$\le \sqrt{\sum_z \phi^2(z)\cdot \sum_z \phi^2(z\oplus \mathsf{Offset})} = \sum_z \phi^2(z) \quad \text{[Cauchy-Schwarz theorem]}$$

$$= \sum_{x,y} \left(\sum_i (-1)^{\mathbb{1}_i\cdot f(x)\oplus \mathbb{1}_i\cdot f(y)} \right)^2$$

$$= \sum_{i,j}\sum_{x,y} (-1)^{\mathbb{1}_i\cdot f(x)\oplus \mathbb{1}_j\cdot f(x)\oplus \mathbb{1}_i\cdot f(y)\oplus \mathbb{1}_j\cdot f(y)}$$

$$= \sum_{i,j} \left(\sum_x (-1)^{\mathbb{1}_i\cdot f(x)\oplus \mathbb{1}_j\cdot f(x)} \right)^2 \quad \left[\begin{array}{c}\text{This is } ||\mathrm{S_4XP}||_2, \textit{ i.e. } \text{Eqn. (F.2)},\\ \text{when } \mathsf{Offset} = \mathtt{0x00}\end{array}\right]$$

$$= m\times 2^{2n} \qquad \begin{cases} m\times(2^n)^2 & \text{if } i=j,\\ 0 & \text{otherwise, because } f \text{ is balanced.}\end{cases}$$

This is proves the result by using $n=6$, $m=4$, and $f=\mathrm{S_4}$.

As a corollary, this noteworthy property of $\mathrm{S_4}$ allows to demonstrate the noting done in [63, §5.1 – pp. 6/7]. It is observed there that:

$$\sum_{\substack{x\in\{0,1\}^6,\\ \mathbb{1}_i\cdot \mathrm{S_4}(x\oplus \mathsf{Offset})=1}} \mathsf{HW}(\mathrm{S_4}(x)) - \sum_{\substack{x\in\{0,1\}^6,\\ \mathbb{1}_i\cdot \mathrm{S_4}(x\oplus \mathsf{Offset})=0}} \mathsf{HW}(\mathrm{S_4}(x)) \tag{F.3}$$

$$= \begin{cases} +32 & \text{if } \mathsf{Offset} = \mathtt{0x00}\,,\\ +32 & \text{if } \mathsf{Offset} = \mathtt{0xf2} \text{ and } i\in\{0,3\}\,,\\ -32 & \text{if } \mathsf{Offset} = \mathtt{0xf2} \text{ and } i\in\{1,2\}\,.\end{cases}$$

Indeed, the expression (F.3), also called "$32\Delta_D$" and noted (11) in [63], can be rewritten as:

$$
\begin{aligned}
& - \sum_{x\in\{0,1\}^6} (-1)^{\mathbb{1}_i\cdot \mathrm{S}_4(x\oplus \mathsf{Offset})} \times \mathsf{HW}(\mathrm{S}_4(x)) \\
= & - \sum_{x\in\{0,1\}^6} (-1)^{\mathbb{1}_i\cdot \mathrm{S}_4(x\oplus \mathsf{Offset})} \left(\sum_{j\in[\![0,3]\!]} \frac{1}{2} - \frac{1}{2}(-1)^{\mathbb{1}_j\cdot \mathrm{S}_4(x)} \right) \\
= & - \frac{1}{2} \left(\underbrace{4 \sum_{x\in\{0,1\}^6} (-1)^{\mathbb{1}_i\cdot \mathrm{S}_4(x\oplus \mathsf{Offset})}}_{\substack{=0, \text{ because} \\ \mathrm{S}_4 \text{ is balanced}}} - \sum_{x\in\{0,1\}^6} \sum_{j\in[\![0,3]\!]} (-1)^{\mathbb{1}_i\cdot \mathrm{S}_4(x\oplus \mathsf{Offset})\oplus \mathbb{1}_j\cdot \mathrm{S}_4(x)} \right) .
\end{aligned}
$$

So, when $\mathsf{Offset} = \mathtt{0x00}$, the expression simplifies in:

$$\frac{1}{2}\sum_x (-1)^0 + \frac{1}{2}\sum_{j\neq i}\sum_x (-1)^{(\mathbb{1}_i\oplus\mathbb{1}_j)\cdot \mathrm{S}_4(x)} = 32 + 0 \ ,$$

since S_4 is balanced.

Besides, when $\mathsf{Offset} = \mathtt{0x2f}$, because of the property of (F.1), the only nonzero cross-term is that for which $i + j = 3$, and it can be seen immediately that it is equal to $\mathrm{S}_4\mathrm{XP}_{i,3-i}/2 = \pm 64/2$, *i.e.* the expected result.

Eventually, the point raised in [63, §5.2 – pp. 7] is also easily explained by Eqn. (F.1). In this context, the leakage model is considered with respect to a reference state $R \in \{0,1\}^4$, and the difference-of-means test yields:

$$
\begin{aligned}
& \sum_{\substack{x\in\{0,1\}^6, \\ \mathbb{1}_i\cdot(\mathrm{S}_4(x\oplus \mathtt{0x2f})\oplus R)=1}} \mathsf{HW}(\mathrm{S}_4(x)\oplus R) - \sum_{\substack{x\in\{0,1\}^6, \\ \mathbb{1}_i\cdot(\mathrm{S}_4(x\oplus \mathtt{0x2f})\oplus R)=0}} \mathsf{HW}(\mathrm{S}_4(x)\oplus R) \quad \text{(F.4)} \\
= \ & \frac{1}{2}\sum_{j=3-i}\sum_x (-1)^{\mathbb{1}_j\cdot \mathrm{S}_4(x)\oplus \mathbb{1}_i\cdot \mathrm{S}_4(x\oplus \mathtt{0x2f})\oplus(\mathbb{1}_j\oplus\mathbb{1}_i)\cdot R} \\
= \ & (-1)^{(\mathbb{1}_{3-i}\oplus\mathbb{1}_i)\cdot R} \times \mathrm{S}_4\mathrm{XP}_{i,3-i}/2 = +32 \text{ if } R = \mathtt{0x4} \ .
\end{aligned}
$$

Coming back to the equiprobable keys in our unrolled implementation, we notice the following corollary derived from the anti-diagonal of Eqn. (F.1):

$$\forall x \in \{0,1\}^6, \mathrm{S}_4(x) \oplus \mathtt{reverse}(\mathrm{S}_4(x \oplus \mathtt{0x2f})) = \mathtt{0x6} \ ,$$

where the function $\mathtt{reverse} : \{0,1\}^4 \to \{0,1\}^4$ swaps bit $i \in [\![0,3]\!]$ with bit $3-i$. Therefore $\forall x, y \in \{0,1\}^6$, $\mathrm{S}_4(x)\oplus\mathrm{S}_4(y)$ and $\mathrm{S}_4(x\oplus\mathtt{0x2f})\oplus\mathrm{S}_4(y\oplus\mathtt{0x2f})$ are palindromic.

Thus, we have a remarkable Hamming distance conservation property: $\forall x, y \in \{0,1\}^6$, $|\mathrm{S}_4(x)\oplus\mathrm{S}_4(y)| = |\mathrm{S}_4(x\oplus\mathtt{0x2f})\oplus\mathrm{S}_4(y\oplus\mathtt{0x2f})|$. As a conclusion, in a Hamming distance model, two keys are retrieved in pairs: the correct one and one another (false), equal to the correct key translated by $\mathtt{0x2f}$.

Appendix G

Formal Analysis of the Entropy / Security Trade-off in First-Order Masking Countermeasures against Side-Channel Attacks

Extended version of article [332]

Authors: *Maxime Nassar, Sylvain Guilley and Jean-Luc Danger*

Abstract

Several types of countermeasures against side-channel attacks are known. The one called masking is of great interest since it can be applied to any protocol and/or algorithm, without nonetheless requiring special care at the implementation level. Masking countermeasures are usually studied with the maximal possible entropy for the masks. However, in practice, this requirement can be viewed as too costly. It is thus relevant to study how the security evolves when the number of mask values decreases. In this chapter, we study a first-order masking scheme, that makes use of one n-bit mask taking values in a strict subset of $\mathbb{F}_2^n$. For a given entropy budget, we show that the security does depend on the choice of the mask values. More specifically, we explore the space of mask sets that resist first- and second-order correlation analysis (CPA and 2O-CPA), using exhaustive search for word size $n \leqslant 5$ bit and a SAT-solver for n up to 8 bit. We notably show that it is possible to protect algorithms against both CPA and 2O-CPA such as AES with only 12 mask values. If the general trend is that more entropy means less leakage, some particular mask subsets can leak less (or on the contrary leak remarkably more). Additionally, we exhibit such mask subsets that allows a minimal leakage.

Keywords: side-channel attacks (SCAs), masking countermeasure, non-injective leakage function, correlation power analysis (CPA), second-order CPA (2O-CPA), mutual information analysis (MIA), entropy *vs* security tradeoff, SAT-solvers.

G.1 Introduction

Implementations of cryptographic algorithms are vulnerable to so-called side-channel attacks. They consist in analysing the leakage of the device during its operation, in a view to relate it to the internal data it processes. The prerequisite of the attack is a physical access to the targeted device. The attacker thus measures some analogue quantity, such as the power [290] or the radiated field [134]. Several ways to resist side-channel have been suggested. They are often referred to as "countermeasures". High level countermeasures intend to deny the exploitation of the leakage by updating the secrets on a regular basis. It results in leakage-resilient protocols. They are nice as they indeed manage to thwart any kind of side-channel attacks, but require that the user adopts a new protocol. Therefore, other countermeasures have been devised that operate at a lower level, without altering the protocol. Typically, hiding strategies aim at leaking a constant side-channel. Although relevant from a theoretical perspective, this approach nonetheless requires physical hypotheses about resources indiscernibility that are not trivial to meet. Masking is another option, that is transparent to the user and does not demand any special backend balance. We therefore focus on this countermeasure. It consists in computing on data whose representation is randomized. The more entropy is used, the more secure the countermeasure can be (if the entropy is used intelligently). In this paper, we rather investigate the effect of the reduction of the entropy on the security. Moreover, we concentrate on a first-order masking scheme, *i.e.* that uses only one mask, that takes a restricted number of values.

The rest of the article is structured as follows. The studied countermeasure, called the rotating tables, is described in Sec. G.2. This section introduces the leakage model considered in the sequel, and defines the notion of leakage and security metrics. The rotating tables countermeasure is then evaluated in the formal framework presented in [434]. Namely, its leakage is characterized in Sec. G.3 and its resistance against CPA and 2O-CPA is quantified in Sec. G.4. It is shown in the section that it is possible to reduce the leakage at a constant budget for masks of $n = 5$ bits. Masks of larger bitwidth, such as $n = 8$, are studied in Sec. G.5. The exploration is conducted with the help of a SAT-solver. Conclusions and perspectives are in Sec. G.6. Some illustrations and long proofs are relegated to appendix.

G.2 Description of the Rotating Tables Countermeasure

The goal of this section is to introduce the leakage model that will be studied next, and to explain why the cost of the countermeasure can be greatly reduced by limiting the mask values. We first give in subsection G.2.1 a brief overview of a masking countermeasure

with randomly selected precomputed tables. Then, in subsection G.2.2, the leakage of this countermeasure is derived.

G.2.1 Rationale

Unprotected implementations are vulnerable to SCAs because they manipulate sensitive variables, that leak some physical quantities that depend somehow on them. Therefore, in a Boolean masking scheme, they are replaced by the exclusive-or (XOR) with random variables. Let us take the example of a first-order masking scheme, where one mask m goes along with one the sensitive variable z. The bitvectors z and m have the same size, namely n bits. We call $\mathcal{S}_0 \doteq z \oplus m$ and $\mathcal{S}_1 \doteq m$ the two shares. The preconditions on the shares is that the sensitive variable can be recovered by XORing them: $Z = \mathcal{S}_0 \oplus \mathcal{S}_1$. The linear operations with respect to the XOR are straightforward. Indeed, to compute a linear operation S on z using the shares, it suffices to apply S on each share. As a matter of fact, it is trivial to check the following post-condition: $S(z) = S(\mathcal{S}_0) \oplus S(\mathcal{S}_1)$. Nonetheless, if S is a non-linear operation, this equality does not hold, and it is necessary to use judiciously both shares to be able to compute $S(z)$. This operation is costly in general [438] (unless some algebraic properties of the non-linear function S can be taken advantage of [383]) and error-prone [295].

Therefore, it is sometimes relevant to compute on only one share, namely $\mathcal{S}_0$. This share traverses the linear parts of the algorithm, and is all-in-one:

1. demasked at the entrance of a non-linear function S,
2. applied S, and
3. remasked so as to propagate through the next linear part.

For sure, the demasking and remasking operations are very sensitive. Nonetheless, the composition of the three operations can be tabulated: a table, such as a ROM block, conceals the intermediate variables (as in whitebox cryptography). Indeed, in cryptography, the non-linear function S will typically be a substitution box (*aka* sbox), that is hard to compute analytically, thus better saved in memory provided there are enough resources to store it. In this case, the intermediate variables never appear. For more details on the implementation of this table, we refer the interested reader to [363, Sec. 2], and more specifically to the paragraphs that concern the "sbox secure calculation".

In a platform that embarks an operating system, a task can be scheduled to recompute the masked sboxes $z \mapsto m_{\text{out}} \oplus S(z \oplus m_{\text{in}})$ periodically. Nonetheless, some embedded systems cannot afford a supervision for the masks update. Also, this process of mask refresh is itself sensitive, and should be protected adequately. In a view to relieve this constraint, one can get rid off the recomputation, and use masked sboxes that had been entered initially. This option is especially favorable for the cryptosystem that reuses several times the same sbox in each round (such as AES). The goal is not to create a security by obscurity solution. Indeed, the masks m_{in} and m_{out} can be disclosed (*i.e.* made public) without compromizing the countermeasure. The randomness that characterizes the masking scheme will result from the choice of the sbox for each computation. Let us take the example of a hardware implementation of AES that computes

one round per clock cycle. Sixteen masked $S[i]$ sboxes, $i \in [\![0, 15]\!]$, must be available in parallel. We assume that the masks $m_{\text{in}}[i]$ and $m_{\text{out}}[i]$ satisfy this chaining relationship: $\forall i \in [\![0, 15]\!], m_{\text{out}}[i] = m_{\text{in}}[i+1 \mod 16]$. Then the computation of an AES-128 can start by drawing a random number $r \in [\![0, 15]\!]$; the algorithm then invokes $S[j + k \mod 16]$ to compute the sbox of byte $j \in [\![0, 16]\!]$ of the state at round $k \in [\![0, 9]\!]$. Because of the chaining property, the linear parts of AES in-between the sboxes are consistently masked and demasked with the same mask. This ensures the correctness of the AES encryption implemented with the rotating sboxes countermeasures.

The overhead of the countermeasure is directly linked to the number of masks[1]. Indeed, more masks mean more memory to store the masked tables. Also, the more tables, the more multiplexing logic to access them, which increases the critical path in a hardware implementation. Thus, in the sequel, we endeavour to reduce the number of masks, while nonetheless keeping an acceptable security level.

G.2.2 Modelization

Hardware implementations of AES are preferably attacked on the last round. Indeed, it is possible to guess one byte, noted y of the round 9 from one byte of the ciphertext x simply by guessing one byte of the last round key, because there is no MixColumns operation in the last round. The leakage is a function of the distance between x and y, *i.e.* $x \oplus y$ [436]. Now, when the rotating tables countermeasure is applied, the value y is actually replaced by $y \oplus m$, where m is one of the 16 mask values. The sensitive variable is the value $x \oplus y$, noted z. In a view to introduce statistical notions, we denote by capital letters (Z and M) the random variables and by small letters (z and m) their realizations. The leakage function thus has the form:

$$\mathcal{L}(Z, M) = \mathscr{L}(Z \oplus M) \ . \tag{G.1}$$

In this expression, Z and M are n-bit vectors, *i.e.* live in $\mathbb{F}_2^n$. The leakage function $\mathscr{L} : \mathbb{F}_2^n \to \mathbb{R}$ depends on the hardware. In a conservative perspective, $\mathscr{L}$ is assumed to be bijective. This choice is the most favorable to the attacker, and is thus considered in the leakage estimation. Now, in practice, the leakage functions are not bijective. The canonical example is that of the Hamming weight leakage, where each bit of $Z \oplus M$ dissipate the same. Let us denote by x_i the component $i \in [\![1, n]\!]$ of $x \in \mathbb{F}_2^n$. The Hamming weight of x is expressed as $\mathsf{HW}(x) = \sum_{i=1}^{n} x_i$.

We underline that this section was not meant to introduce a new countermeasure (the rotating sboxes). Indeed, this pragmatic countermeasure is already well known and adopted in the industry [178, 334]. We simply wished to provide the reader with a pedagogical introduction to the leakage function of Eqn. (G.1). This function will now be studied formally, as per the guidelines presented in [434]. More precisely, we employ:

1. Notice that in the rest of the article, we have only one masking variable, that takes few values. We sometimes refer to them as the "number of masks"; we attract the reader's attention on the fact this expression shall not be confused with "multi-masks" countermeasures, also known as "high-order" masking schemes.

– The mutual information between the $\mathcal{L}(Z, M)$ and the sensitive variable Z with $\mathscr{L}$ bijective as a leakage metric. This quantity is noted $\mathsf{I}[\mathcal{L}(Z, M); Z]$ — basic definitions of information theory applied to SCAs can be found in [434] — and referred to as "mutual information as a metric" (MIM [468]). We recall that a leakage metric points out vulnerabilities, that could in practice not be exploited by an attacker.
– Security metrics to quantify the easiness to actually turn a leakage into a successful attack. In this case, we will focus on $\mathscr{L} = \mathsf{HW}$. First of all, the optimal correlation between $\mathsf{HW}(Z \oplus M)$ and Z is considered a metric. It is traditionally called the (first-order) correlation power analysis, or CPA [60]. But CPA can be defeated easily with only two mask values. Therefore it is important to consider higher-order CPA (HO-CPA), and notably the second-order CPA, also abridged 2O-CPA [470]. However, CPA and 2O-CPA exploit only the first two moments of the distribution of $\mathcal{L}(Z, M)$. Therefore, we also use a second security metric, namely the mutual information. It is known in the literature as MIA [23]. Security-wise, our goal is to minimize the first- and second-order correlation coefficients and the MIA.

G.3 Information Theoretic Evaluation of the Countermeasure

The specificity of this study is to consider masks M that are not completely entropic. Thus, the probability $\mathsf{P}[M = m]$ depends on m. Our target is to restrict to a relevant subset of the masks uniformly, that is every mask is used with the same probability. We call $\mathcal{M} \subseteq \mathbb{F}_2^n$ the set of masks actually used. Thus:

$$\mathsf{P}[M = m] = \begin{cases} 1/\mathsf{Card}[\mathcal{M}] & \text{if } m \in \mathcal{M}, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

We also write this probability law $M \sim \mathcal{U}(\mathcal{M})$. From an information theoretic point of view, we can characterize the entropy of M. By definition,

$$\mathsf{H}[M] = -\sum_{m \in \mathcal{M}} \frac{1}{\mathsf{Card}[\mathcal{M}]} \log_2 \frac{1}{\mathsf{Card}[\mathcal{M}]} = \log_2 \mathsf{Card}[\mathcal{M}] \text{ bit}\,.$$

The minimal number of masks is 1, which corresponds to the absence of countermeasure (take $M = 0$ in Eqn. (G.1)). At the opposite, when all the 2^n masks are used, the countermeasure is optimal.

Eventually, we assume that the attacker does not conduct a chosen message attack, *i.e.* $Z \sim \mathcal{U}(\mathbb{F}_2^n)$. We notice that even if the attacker cannot actually choose the messages, she has nonetheless the possibility to discard some messages so as to artificially bias the side-channel attack. But a priori, the attacker does not know which plaintext Z to favor. A biased side-channel attack has been detailed in [252, 469]. However, this attack is adaptative, and thus requires that a breach be already found. Nonetheless, in our context, we target the protection of the secret at the early stages of the attack; the

attacker still does not have any clue about the most likely hypotheses for the secret. This hypothesis is called the *non-adaptive known plaintext model* in [434].

Whatever the actual leakage function $\mathscr{L}$, $\mathsf{I}[\mathscr{L}(Z \oplus M); Z] = 0$ if $\mathsf{H}[M] = n$ bit (or equivalently, if $M \sim \mathcal{U}(\mathbb{F}_2^n)$). So with all the masks, the countermeasure is perfect.

If $\mathscr{L}$ is bijective (*e.g.* $\mathscr{L} = \mathsf{Id}$), then $\mathsf{I}[\mathscr{L}(Z \oplus M); Z] = n - \mathsf{H}[M]$. This results directly from the observation that:

- $\mathsf{H}[\mathscr{L}(Z \oplus M)] = \mathsf{H}[\mathscr{L}(Z)] = n$ bit, since $Z \sim \mathcal{U}(\mathbb{F}_2^n)$, and
- $\mathsf{H}[\mathscr{L}(Z \oplus M) \mid Z] = \mathsf{H}[M]$ bit because Z and M are independent.

We notice that this quantity is independent of the exact $\mathcal{M}$, provided $\mathsf{Card}[\mathcal{M}]$ is fixed. This means that degrading the countermeasure (*i.e.* choosing $\mathsf{Card}[\mathcal{M}] < 2^n$) introduces a vulnerability, while decreasing the cost.

Now, it can checked to which extent this vulnerability is exploitable, considering a realistic leakage function. Specifically, it can be shown that if $\mathscr{L}$ is not injective, then the MIA metric $\mathsf{I}[\mathscr{L}(Z \oplus M); Z]$ depends on $\mathcal{M}$. Appendix G.7 provides with an example. More precisely, when $\mathcal{M}$ as two (complementary) elements, then the MIA is independent of $\mathcal{M}$ (refer to appendix G.8). But when $\mathcal{M}$ is made up of strictly more than two masks, the MIA depends on $\mathcal{M}$. For example, on $n = 8$ bits,

- $\mathsf{I}[\mathscr{L}(Z \oplus M); Z] = 1.42701$ bit if $\mathcal{M} = \{\texttt{0x00}, \texttt{0x0f}, \texttt{0xf0}, \texttt{0xff}\}$, but
- $\mathsf{I}[\mathscr{L}(Z \oplus M); Z] = 0.73733$ bit if $\mathcal{M} = \{\texttt{0x00}, \texttt{0x01}, \texttt{0xfe}, \texttt{0xff}\}$.

Thus, it is relevant to search for mask sets, at a constant budget (*i.e.* for a given $\mathsf{Card}[\mathcal{M}]$), that minimize the mutual information $\mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$. Nonetheless, without a method, it is not obvious to conduct a reasoned search. Indeed, the default solution is to draw at random one mask set $\mathcal{M}$ and to compute $\mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$. It is immediate to see that such method will indeed provide solutions harder to attack using MIA than the others, but that will maybe fail in front of other less sophisticated attacks. Typically, $\mathcal{M}$ sets only constrained by their cardinality are likely to yield functions trivially attackable by CPA. We therefore propose the following method:

- First mask sets $\mathcal{M}$ that resist first- and second order correlation attacks (*i.e.* CPA and 2O-CPA, the easiest attacks against single-masked countermeasures) are found. This is the topic of Sec. G.4.
- Then, amongst these solutions, those minimizing the risk of MIA are selected. Section G.5 specifically analyses this point (already quickly discussed in Sec. G.4.5).

Another argument to focus primarily on CPA and 2O-CPA is that they require in practice less side-channel measurements to succeed the attack than MIA. Indeed, MIA, as well all other information theoretic-based attacks (*e.g.* template attacks [69] and stochastic attacks [406]), need to estimate conditional probability functions, which needs many traces [141]. Also, from the certification standpoint, the common criteria [1] demand that the implemented countermeasures resist "state-of-the-art" attacks [91]. Now, CPA and 2O-CPA are much more studied in the information technology security evaluation facilities (ITSEFs) than information theoretic attacks.

G.4 Security against CPA and 2O-CPA

The average of the leakage function given in Eqn. (G.1) depends on $\mathscr{L} : \mathbb{F}_2^n \to \mathbb{R}$. As already mentioned, to conduct exact computations and to match with realistic leakage functions observed in practice, we opt for the Hamming weight ($\mathscr{L} = \mathsf{HW}$). Thus the average of leakage function, noted $\mathsf{E}\mathcal{L}(Z, M)$, is equal to:

$$\mathsf{E}\ \mathsf{HW}(Z \oplus M) = \frac{1}{\mathsf{Card}[\mathcal{M}]} \sum_{m \in \mathcal{M}} \frac{1}{2^n} \sum_{z \in \mathbb{F}_2^n} \mathsf{HW}(z \oplus m) = \frac{1}{\mathsf{Card}[\mathcal{M}]} \sum_{m \in \mathcal{M}} \frac{n}{2} = \frac{n}{2}. \quad \text{(G.2)}$$

Against HO-CPA of order $d \geqslant 1$, the most powerful attacker correlates her guesses about the sensitive variable with the optimal function [365] defined as:

$$\begin{aligned} f_{\text{opt}}^{(d)}(z) &\doteq \mathsf{E}\left((\mathcal{L}(Z, M) - \mathsf{E}\mathcal{L}(Z, M))^d \mid Z = z \right) \\ &= \mathsf{E}\left(\left(\mathsf{HW}(Z \oplus M) - \frac{n}{2} \right)^d \mid Z = z \right) \\ &= \frac{1}{\mathsf{Card}[\mathcal{M}]} \sum_{m \in \mathcal{M}} \left(\frac{-1}{2} \sum_{i=1}^{n} (-1)^{(z \oplus m)_i} \right)^d, \end{aligned} \quad \text{(G.3)}$$

because if $b \in \{0, 1\}$, then $b - \frac{1}{2} = -\frac{1}{2}(-1)^b$. Recall that the rotating tables countermeasure uses only one mask variable M, and thus leaks at only one date (*i.e.* for a given timing sample). In this context, HO-CPA consists in studying the linear dependency between the d-th moments of the leakage classes and the optimal function $f_{\text{opt}}^{(d)}(z)$ of the sensitive variable z.

For the designer of the countermeasure, the objective is to make Eqn. (G.3) independent of z. There is always a solution that consists in choosing $\mathcal{M} = \mathbb{F}_2^n$. Nonetheless, with $\mathsf{Card}[\mathcal{M}] < 2^n$, the existence of solutions is a priori not trivial. In this case, if is impossible to find masks that keep $f_{\text{opt}}^{(d)}(z)$ (defined in Eqn. (G.3)) independent from z, the secondary goal is to minimize the correlation coefficient:

$$\rho_{\text{opt}}^{(d)} \doteq \frac{\mathsf{Var}\left(f_{\text{opt}}^{(d)}(Z) \right)}{\mathsf{Var}\left((\mathcal{L}(Z, M) - \mathsf{E}\mathcal{L}(Z, M))^d \right)} = \frac{\mathsf{Var}\left(\mathsf{E}\left(\left(\mathsf{HW}(Z \oplus M) - \frac{n}{2} \right)^d \mid Z \right) \right)}{\mathsf{Var}\left(\left(\mathsf{HW}(Z \oplus M) - \frac{n}{2} \right)^d \right)}. \quad \text{(G.4)}$$

In this equation, Var represents the variance, defined on a random variable X as $\mathsf{Var}(X) \doteq \mathsf{E}\,(X - \mathsf{E}X)^2$.

In the two next subsections G.4.1 and G.4.2, the analytical expression of Eqn. (G.4) is derived. Then these expressions are unified in subsection G.4.3 by replacing the notion of subset $\mathcal{M}$ by an indicator function f. The sets of masks that completely allow to deny CPA and 2O-CPA are given exhaustively in subsection G.4.4 for $n = 4$ and in subsection G.4.5 for $n = 5$.

G.4.1 Resistance against First-Order Correlation Attacks

As shown in appendix G.9.1, when $d = 1$, Eqn. (G.4) is equal to:

$$\rho_{\text{opt}}^{(1)} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{\mathsf{Card}[\mathcal{M}]} \sum_{m \in \mathcal{M}} (-1)^{m_i} \right)^2 . \tag{G.5}$$

This correlation $\rho_{\text{opt}}^{(1)}$ can be equal to zero if and only if (iff), for all $i \in [\![1, n]\!]$, $\mathsf{E}M_i = 1/2$. This means that the masks are balanced. It is possible to find such masks iff $\mathsf{Card}[\mathcal{M}]$ is a multiple of two. A construction consists in building a set of masks by adding a new mask and its complement. Conversely, in a set containing an odd number of different masks, it is impossible to as many ones as zeros for any component. For instance, we illustrate how to generate balanced sets of masks in the case $n = 4$ in Tab. G.1.

A trivial example consists in taking two masks, m and $\neg m$ (such as `0x00` and `0xff` on $n = 8$ bits). This is sufficient to thwart first-order attacks. At the opposite, without mask ($\mathcal{M}$ is equal to the singleton $\{\texttt{0x00}\}$) or with a single mask ($\mathcal{M} = \{m\}$, whatever $m \in \mathbb{F}_2^n$), the correlation coefficient reaches its maximum (*i.e.* $+1$, because Eqn. (G.4) considers a correlation in absolute value).

G.4.2 Resistance against Second-Order Correlation Attacks

As shown in appendix G.9.2, when $d = 2$, Eqn. (G.4) is equal to:

$$\rho_{\text{opt}}^{(2)} = \frac{1}{n(n-1)} \left(\frac{1}{\mathsf{Card}[\mathcal{M}]^2} \sum_{(m,m') \in \mathcal{M}^2} \left(\sum_{i=1}^{n} (-1)^{(m \oplus m')_i} \right)^2 - n \right) . \tag{G.6}$$

As an illustration, we show in Tab. G.2 the optimal correlation coefficients of order 1 and 2 for the masks sets of Tab. G.1 ($n = 4$ bit). We have added a column (the last one), for $\mathscr{L} = \mathsf{Id}$; also, in the last row, we have included a constant masking (unprotected implementation), which serves as a reference.

G.4.3 Expression of $\rho_{\text{opt}}^{(1,2)}$ as a Function of an Indicator f

The expressions of $\rho_{\text{opt}}^{(1)}$ and $\rho_{\text{opt}}^{(2)}$ (altogether referred to as $\rho_{\text{opt}}^{(1,2)}$) defined in Eqn. (G.5) and (G.6) lay a mathematical ground to search for suitable $\mathcal{M}$. Nonetheless, these equations remain at the set-theory level. To simplify the problem, we introduce the Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, defined as: $\forall m \in \mathbb{F}_2^n, f(m) = 1 \iff m \in \mathcal{M}$. Then, we can simply replace "$\sum_{m \in \mathcal{M}}$" by "$\sum_{m \in \mathbb{F}_2^n} f(m)$" in the equations previously established.

The Fourier transform $\hat{f} : \mathbb{F}_2^n \to \mathbb{Z}$ of the Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined as $\forall a \in \mathbb{F}_2^n, \hat{f}(a) \doteq \sum_{m \in \mathbb{F}_2^n} f(m)(-1)^{a \cdot m}$. It allows for instance to write $\mathsf{Card}[\mathcal{M}] = \sum_{m \in \mathcal{M}} 1 = \sum_{m \in \mathbb{F}_2^n} f(m) = \hat{f}(0)$. Recall $\mathsf{Card}[\mathcal{M}] \in [\![1, 2^n]\!]$, hence $\hat{f}(0) > 0$.

Table G.1: Mask sets $\mathcal{M}$ that make the masking countermeasure immune to first order CPA. The masks go by pair, symmetrically with the middle of the table.

	$\mathsf{Card}[\mathcal{M}] = 2^4$	$\mathsf{Card}[\mathcal{M}] = 2^3$	$\mathsf{Card}[\mathcal{M}] = 2^2$	$\mathsf{Card}[\mathcal{M}] = 2^1$
	0000	0000	0000	0000
	0001			
	0010			
	0011	0011	0011	
	0100	0100		
	0101			
	0110			
$\mathcal{M}$	0111	0111		
	1000	1000		
	1001			
	1010			
	1011	1011		
	1100	1100	1100	
	1101			
	1110			
	1111	1111	1111	1111

Table G.2: Security metrics for the masks sets of Tab. G.1 and the singleton.

$\mathsf{Card}[\mathcal{M}]$	$\mathsf{H}[M]$	$\rho^{(1)}_{\text{opt}}$	$\rho^{(2)}_{\text{opt}}$	$\mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$	$\mathsf{I}[Z \oplus M; Z]$
2^4	4	0	0	0	0
2^3	3	0	0.166667	0.15564	1
2^2	2	0	0.333333	1.15564	2
2^1	1	0	1	1.40564	3
2^0	0	1	1	2.03064	4

Then Eqn. (G.5) rewrites:

$$\rho_{\text{opt}}^{(1)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{f}(e_i)}{\hat{f}(0)}\right)^2 , \tag{G.7}$$

where e_i are the canonical basis vectors $(0,\cdots,0,1,0,\cdots,0)$, the unique 1 laying at position i.

Also, Eqn. (G.6) rewrites:

$$\begin{aligned}\rho_{\text{opt}}^{(2)} &= \frac{1}{n(n-1)}\sum_{(i,i')\in[\![1,n]\!]^2}\left(\left(\frac{\hat{f}(e_i\oplus e_{i'})}{\hat{f}(0)}\right)^2 - n\right)\\ &= \frac{1}{n(n-1)}\sum_{\substack{(i,i')\in[\![1,n]\!]^2\\ i\neq i'}}\left(\frac{\hat{f}(e_i\oplus e_{i'})}{\hat{f}(0)}\right)^2 .\end{aligned} \tag{G.8}$$

Thus, the rotating tables countermeasure resists:

1. first-order attacks iff $\forall a, \mathsf{HW}(a)=1 \Longrightarrow \hat{f}(a)=0$;
2. first- and second-order attacks iff $\forall a, 1\leqslant \mathsf{HW}(a)\leqslant 2 \Longrightarrow \hat{f}(a)=0$.

As a sanity check, we can verify that these properties hold when all the 2^n masks are used, *i.e.* when f is constant (and furthermore equal to 1). Indeed, in this case, $\hat{f}(a) = \sum_m f(m)(-1)^{a\cdot m} = \sum_m (-1)^{a\cdot m} = 2^n\delta(a)$, where δ is the Kronecker symbol.

Now, we notice that for Boolean functions, the notions of Fourier and Walsh transforms are very alike. Indeed,

$$\forall a\neq 0, \hat{f}(a) = \textstyle\sum_m f(a)(-1)^{a\cdot m} = \sum_m (-1)^{a\cdot m}\frac{1}{2}\left(1-(-1)^{f(m)}\right) = -\frac{1}{2}\widehat{(-1)^f}(a)\,.$$

Therefore, the previous conditions are equivalent to saying the following: the countermeasure resists $d\in\{1,2\}$ order CPA iff $\forall a, \mathsf{HW}(a)\leqslant d \Longrightarrow \widehat{(-1)^f}(a)=0$.

We insist that this characterization is <u>not</u> equivalent to saying that f is d-resilient (defined in [65, page 45]). Indeed, a resilient function is balanced, which is explicitly not the case of f. Therefore, we study in the sequel a new kind of Boolean functions, that have everything in common with resilient functions but the balancedness of the plain function. The corollary is that, to the authors' best knowledge, no known construction method exists for this type of functions. Nonetheless, it is interesting to get an intuition about what characterizes a good resilient function. In [65, §7.1, page 95], it is explained that the highest degree of resiliency of a $f:\mathbb{F}_2^n\to\mathbb{F}_2$ is $n-2$. This maximum is reached by affine functions (functions of unitary algebraic degree). Nonetheless, in our case, affine functions are not the best choice, because they are balanced. This means that the cardinality of their support (*i.e.* $\mathsf{Card}[\mathcal{M}]$) is 2^{n-1}, which is large. Therefore, we will be interested, whenever possible, by non-affine functions f of algebraic degree strictly greater than one (noted $d^\circ_{\text{alg}}(f)>1$).

Table G.3: All the functions $f : \mathbb{F}_2^4 \to \mathbb{F}_2$ that cancel $\rho_{\text{opt}}^{(1,2)}$.

f	$\mathsf{HW}(f)$	$\mathsf{H}[M]$	$\rho_{\text{opt}}^{(1)}$	$\rho_{\text{opt}}^{(2)}$	$\mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$	$\mathsf{I}[Z \oplus M; Z]$	$d_{\text{alg}}^{\circ}(f)$
0x3cc3	8	3	0	0	0.219361	1	1
0x5aa5	8	3	0	0	0.219361	1	1
0x6699	8	3	0	0	0.219361	1	1
0x6969	8	3	0	0	0.219361	1	1
0x6996	8	3	0	0	1	1	1
0x9669	8	3	0	0	1	1	1
0x9696	8	3	0	0	0.219361	1	1
0x9966	8	3	0	0	0.219361	1	1
0xa55a	8	3	0	0	0.219361	1	1
0xc33c	8	3	0	0	0.219361	1	1
0xffff	16	4	0	0	0	0	0

G.4.4 Functions $f : \mathbb{F}_2^4 \to \mathbb{F}_2$ that Cancel $\rho_{\text{opt}}^{(1,2)}$

For $n = 4$, all the sets $\mathcal{M}$ can be tested. The table G.3 reports all the functions f that cancel $\rho_{\text{opt}}^{(1)}$ and $\rho_{\text{opt}}^{(2)}$. In this table, the truth-table of f, given in the first column, is encoded in hexadecimal. We note $\mathsf{HW}(f)$ the number of ones in the truth-table, and recall that $\mathsf{HW}(f) = \mathsf{Card}[\mathcal{M}]$. Columns 4, 5 and 6 are security metrics, whereas column 7 is the leakage metric (MIM). There are non-trivial solutions only for $\mathsf{Card}[\mathcal{M}]$ equal to half of the complete mask set cardinal. The MIA (column 6) shows two values: 0.219361 and 1 bit. Those values shall be contrasted with the MIA:

- without countermeasure ($\mathsf{Card}[\mathcal{M}] = 1$): MIA = 2.19819 bit and
- with two complementary masks ($\mathsf{Card}[\mathcal{M}] = 2$, which thwarts CPA but not 2O-CPA): MIA = 1.1981 bit (refer to appendix G.8).

Thus the countermeasure resists better correlation and information theoretic attacks, at the expense of more masks. Indeed, apart from $f = 1$, all the solutions are affine ($d_{\text{alg}}^{\circ}(f) = 1$), and thus have a Hamming weight of $2^{n-1} = 8 \gg 2$.

In this table, some functions belong to equivalent classes. Namely, two of them can be identified:

- the permutations of the bits (because the summations over i in Eqn. (G.7) or i, i' in Eqn. (G.8) is invariant in any change of the bits order), and
- the complementation. Indeed, $\widehat{\neg f}(a) = \sum_{m \in \mathbb{F}_2^n} \neg f(m)(-1)^{a \cdot m} = \sum_{m \in \mathbb{F}_2^n} (1 - f(m))(-1)^{a \cdot m} = 2^n \delta(a) - \hat{f}(a)$. Now, in Eqn. (G.7) and (G.8), $a \neq 0$ and $\hat{f}$ is involved squared. Thus $\rho_{\text{opt}}^{(1,2)}(\neg f) = \rho_{\text{opt}}^{(1,2)}(f)$.

The same can be said for the mutual information. This lemma is useful:

Lemma 1. *Let A and B be two random variables and ϕ a bijection;*

then $\mathsf{I}[A;\phi(B)] = \mathsf{I}[A;B]$.

This equality is obtained simply by writing the definition of the mutual information as a function of the probabilities, and by doing a variable change. Then:

- Let us call σ a permutation of $[\![1,n]\!]$. This function is a bijection, and its inverse is also a permutation. The Hamming weight is invariant if σ is applied on its input (*i.e.* $\mathsf{HW} = \mathsf{HW} \circ \sigma$). Hence $\mathsf{HW}(Z \oplus \sigma(M)) = \mathsf{HW}(\sigma^{-1}(Z \oplus \sigma(M))) = \mathsf{HW}(\sigma^{-1}(Z) \oplus M)$ (because σ is furthermore linear with respect to the addition). Let us note $Z' = \sigma^{-1}(Z)$, a random variable that is also uniform. Thus, $\mathsf{I}[\mathsf{HW}(Z \oplus \sigma(M)); Z] = \mathsf{I}[\mathsf{HW}(Z' \oplus M); \sigma(Z')]$. By considering $\phi = \sigma$, we prove that $\mathsf{I}[\mathsf{HW}(Z \oplus \sigma(M)); Z] = \mathsf{I}[\mathsf{HW}(Z' \oplus M); Z'] = \mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$, because Z and Z' have the same probability density function.
- Regarding the complementation, it is straightforward to note that $\mathsf{HW}(Z \oplus \neg M) = \mathsf{HW}(\neg(Z \oplus M)) = n - \mathsf{HW}(Z \oplus M)$. By considering $\phi : x \mapsto n - x$, we also have the invariance of the mutual information by the complementation of the mask.

So, there are eventually only three classes of functions listed in Tab. G.3, modulo the two abovementioned equivalence classes. They are summarized below:

1. $f(x_1, x_2, x_3, x_4) = \bigoplus_{\substack{i \in I \subseteq [\![1,4]\!] \\ \mathsf{Card}[I]=3}} x_i$, (*aka* `0x3cc3`, `0x5aa5`, `0x6699`, `0x6969`) or complemented (*aka* `0x9696`, `0x9966`, `0xa55a`, `0xc33c`); According to the criteria stated at the end of Sec. G.3, those functions are the best solutions for $n = 4$.
2. $f(x_1, x_2, x_3, x_4) = \bigoplus_{i=1}^{4} x_i$ (*aka* `0x6996`) or $f(x_1, x_2, x_3, x_4) = 1 \oplus \bigoplus_{i=1}^{4} x_i$ (*aka* `0x9669`), that have no advantage over the previous solutions;
3. the constant function $f = 1$ (*aka* `0xffff`).

To resist first-order attacks, the masks set can be partitioned in two complementary sets; this means that there exists $\tilde{\mathcal{M}}$, a subset of $\mathcal{M}$, such that: $\mathcal{M} = \tilde{\mathcal{M}} \cup \neg\tilde{\mathcal{M}}$, where $\neg\tilde{\mathcal{M}} \doteq \{\neg m, m \in \mathcal{M}\}$. Incidentally, we notice that this is not a mandatory property. Typically, this property is not verified any longer at order 2. For instance, in the solution $f = \texttt{0x3cc3}$, $\texttt{0x0} \in \mathcal{M}$ but $\neg\texttt{0x0} = \texttt{0xf} \notin \mathcal{M}$.

In conclusion, when $n = 4$ and the designer cannot afford using all the 16 masks, then with 8 masks, the rotating tables countermeasure is able to resist CPA, 2O-CPA and leak the minimal value of 0.219361 bit (about ten times less than the unprotected implementation, for which the MIA is 2.19819 bit).

G.4.5 Functions $f : \mathbb{F}_2^5 \to \mathbb{F}_2$ that Cancel $\rho_{\mathbf{opt}}^{(1,2)}$

For $n = 5$, all the subsets $\mathcal{M}$ of $\mathbb{F}_2^5$ (2^{32} of them, it is the maximum achievable on a personal computer, as precised in [65, page 6]) have been tested. There are 1057 functions that cancel $\rho_{\text{opt}}^{(1,2)}$. The lowest value for $\mathsf{HW}(f)$ is 8. There are 60 functions of weight 8, but only three classes modulo the invariants. The functions, sorted regarding their properties, are shown in Tab. G.4. As opposed to the case $n = 4$, there are non-affine solutions. In this table, only the number of equivalent classes is given. For a list of all functions, refer to appendix G.10.1.

Table G.4: Summary of the security metrics of $f : \mathbb{F}_2^5 \to \mathbb{F}_2$ that cancel $\rho_{\text{opt}}^{(1,2)}$.

Nb. classes	$\mathsf{HW}(f)$	$\mathsf{H}[M]$	$\rho_{\text{opt}}^{(1)}$	$\rho_{\text{opt}}^{(2)}$	$\mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$	$\mathsf{I}[Z \oplus M; Z]$	$d_{\text{alg}}^{\circ}(f)$
3	8	3	0	0	0.32319	2	2
4	12	3.58496	0	0	0.18595	1.41504	3
2	16	4	0	0	0.08973	1	1
2	16	4	0	0	0.08973	1	2
4	16	4	0	0	0.12864	1	2
2	16	4	0	0	0.16755	1	1
4	16	4	0	0	0.26855	1	2
6	16	4	0	0	0.32495	1	2
1	16	4	0	0	1	1	1
4	20	4.32193	0	0	0.07349	0.67807	3
3	24	4.58496	0	0	0.04300	0.41504	2
1	32	5	0	0	0	0	0

The greater $\mathsf{H}[M]$, the smaller the mutual information with $\mathscr{L} = \mathsf{HW}$ in general, but for some remarkable solutions (*e.g.* the one MIA $= \mathsf{I}[\mathsf{HW}(Z \oplus M); Z] = 1$ of algebraic degree 1 for $\mathsf{HW}(f) = 16$). Also, it is worth noting that for a given budget (*e.g.* 16 masks) and security requirement (resistance against CPA and 2O-CPA), some solutions are better than the others against MIA. Indeed, the leaked information in Hamming weight model spans from 0.0897338 bit to 1 bit.

G.5 Exploring More Solutions Using SAT-Solvers

In order to explore problems of greater complexity, SAT-solver are indicated tools. We model f as a set of 2^n Boolean unknowns. The problem consists in finding f such that $\forall a, 1 \leqslant \mathsf{HW}(a) \leqslant 2, \hat{f}(a) = 0$, for a given $\mathsf{Card}[\mathcal{M}] = \hat{f}(0)$. A SAT-solver either:

- proves that there is no solution, or
- proves that a solution exists, and provides for (at least) one.

We notice that a SAT-solver may not terminate on certain instances of large exploration space; this has not been an issue in the work we report here. In this section, we first explain how our problem can be fed into a SAT-solver. Then, we use a SAT-solver in the case $n = 8$, relevant for AES. We look for low $\mathsf{Card}[\mathcal{M}]$ solutions, and for a given $\mathsf{Card}[\mathcal{M}]$, for the solutions of minimal MIA.

G.5.1 Mapping of the Problem into a SAT-Solver

Knowing that $\mathsf{Card}[\mathcal{M}] = \hat{f}(0)$, the problem $\rho^{(1,2)}_{\text{opt}}(f) = 0$ rewrites:

$$\forall a, 1 \leqslant \mathsf{HW}(a) \leqslant 2, \quad \sum_x f(x)(-1)^{a\cdot x} = 0 \quad \Longleftrightarrow$$

$$\forall a, 1 \leqslant \mathsf{HW}(a) \leqslant 2, \quad \sum_x f(x) \wedge (a \cdot x) = \frac{1}{2}\sum_x f(x) = \frac{1}{2}\mathsf{Card}[\mathcal{M}] \ . \quad \text{(G.9)}$$

A SAT-solver verifies the validity of clauses, usually expressed in conjunctive normal form (CNF). It is known that cardinality constraints can be formulated compactly thanks to Boolean clauses. More precisely, any condition "$\leqslant k(x_1, \cdots, x_n)$", for $0 \leqslant k \leqslant n$, can be expressed in terms of CNF clauses [415]. We note that:

$$\mathsf{HW}(x) \leqslant k \quad \Longleftrightarrow \quad n - \mathsf{HW}(\neg x) \leqslant k \quad \Longleftrightarrow \quad \mathsf{HW}(\neg x) \geqslant n - k\,.$$

Hence, satisfying $\geqslant k(x_1, \cdots, x_n)$ is equivalent to satisfying $\leqslant n - k(\neg x_1, \cdots, \neg x_n)$. Thus, testing the equality of a Hamming to $\frac{1}{2}\mathsf{Card}[\mathcal{M}]$ can be achieved by the conjunction of two clauses: $\leqslant \frac{1}{2}\mathsf{Card}[\mathcal{M}](x_1, \cdots, x_n)$ and $\leqslant n - \frac{1}{2}\mathsf{Card}[\mathcal{M}](\neg x_1, \cdots, \neg x_n)$.

The $n = 8$, the number of useful literals, $\{f(x), x \in \mathbb{F}_2^n\}$, is 2^8. However, the constraints $\mathsf{Card}[\mathcal{M}] = \hat{f}(0)$ and $\rho^{(1,2)}_{\text{opt}}(f) = 0$ (see Eqn. (G.9)) introduce 1,105,664 auxiliary variables and translate into 2,219,646 clauses, irrespective of $\mathsf{Card}[\mathcal{M}] \in \mathbb{N}^*$.

G.5.2 Existence of Low Hamming Weight Solutions for $n = 8$

The software `cryptominisat` [421, 422] is used to search for solutions. The problem is tested for all the $\mathsf{Card}[\mathcal{M}]$ from 2 to 2^n, by steps of 2, as independent problems. Each problem requires a few hours to be solved. Impressively low Hamming weight solutions are found. The table G.5 represents some of them. There are solutions only for $\mathsf{Card}[\mathcal{M}] \in \{4 \times \kappa, \kappa \in [\![3, 61]\!] \cup \{64\}\}$. Also, the mutual information with a Hamming weight leakage as a function of $\mathsf{H}[M]$ is plotted in Fig. G.1. These values are low when compared to:

- MIA = 2.5442 bit without masking ($\mathsf{Card}[\mathcal{M}] = 1$) and
- MIA = 1.8176 bit with a mask that takes two complementary values ($\mathsf{Card}[\mathcal{M}] = 2$).

Those MIA figures are computed in appendix G.8, and concern countermeasures that do not protect against 2O-DPA. The table G.5 basically indicates that the margin gain in MIA resistance decreases when the cost of the countermeasures, proportional to $\mathsf{HW}(f)$, increases.

G.5.3 Exploration of Solutions for $n = 8$ and a Fixed $\mathsf{Card}[\mathcal{M}]$

There are nonequivalent solutions for a same $\mathsf{Card}[\mathcal{M}]$. Various seeds of the SAT-solver are needed to discover these solutions. The appendix G.10.2 gives some nonequivalent solutions for the minimal value $\mathsf{Card}[\mathcal{M}] = 12$, and details the truth-table of

Table G.5: Metrics for one $f : \mathbb{F}_2^8 \to \mathbb{F}_2$ (per support cardinality) that cancels $\rho_{\text{opt}}^{(1,2)}$, found by a SAT-solver.

$\mathsf{HW}(f)$	$\mathsf{H}[M]$	$\rho_{\text{opt}}^{(1)}$	$\rho_{\text{opt}}^{(2)}$	$\mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$	$\mathsf{I}[Z \oplus M; Z]$	$d_{\text{alg}}^{\circ}(f)$
12	3.58496	0	0	0.387582	4.41504	6
16	4	0	0	0.219567	4	5
20	4.32193	0	0	0.228925	3.67807	6
24	4.58496	0	0	0.235559	3.41504	5
28	4.80735	0	0	0.144147	3.19265	6
32	5	0	0	0.135458	3	5
36	5.16993	0	0	0.090575	2.83007	6
40	5.32193	0	0	0.078709	2.67807	5
44	5.45943	0	0	0.067960	2.54057	6
48	5.58496	0	0	0.060515	2.41504	5
52	5.70044	0	0	0.092676	2.29956	6
56	5.80735	0	0	0.054936	2.19265	5
60	5.90689	0	0	0.049069	2.09311	6
64	6	0	0	0.035394	2	2
68	6.08746	0	0	0.042374	1.91254	6
72	6.16993	0	0	0.036133	1.83007	5
76	6.24793	0	0	0.034194	1.75207	6
80	6.32193	0	0	0.031568	1.67807	5
84	6.39232	0	0	0.030072	1.60768	6
88	6.45943	0	0	0.026941	1.54057	5
92	6.52356	0	0	0.027042	1.47644	6
96	6.58496	0	0	0.022992	1.41504	5
100	6.64386	0	0	0.024316	1.35614	6
104	6.70044	0	0	0.022257	1.29956	5
108	6.75489	0	0	0.021458	1.24511	6
112	6.80735	0	0	0.019972	1.19265	4
116	6.85798	0	0	0.020481	1.14202	6
120	6.90689	0	0	0.018051	1.09311	5
124	6.9542	0	0	0.018397	1.0458	6
128	7	0	0	0.015095	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮

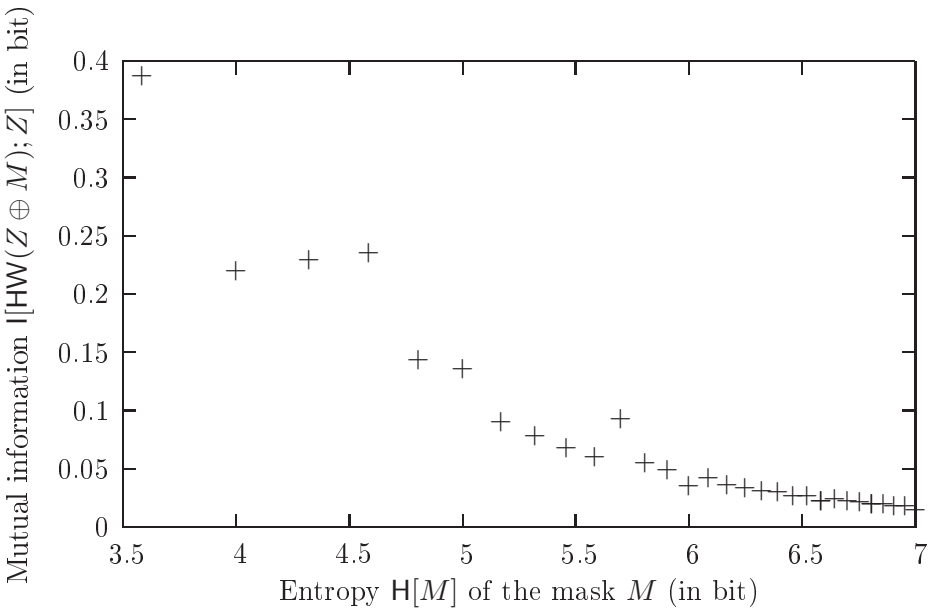


Figure G.1: Mutual information of the leakage in Hamming weight with the sensitive variable Z, for one solution that cancels $\rho_{\text{opt}}^{(1,2)}$ found by the SAT-solver.

one solution. All the solutions found by the SAT-solver for $\mathsf{Card}[\mathcal{M}] = 12$ have the same MIA value: 0.387582 bit. The same section in the appendix shows that for $\mathsf{Card}[\mathcal{M}] = 16$, various MIA values exist. The SAT-solver has notably came across, from best to worst: 0.181675, 0.213996, 0.215616, 0.216782, 0.219567, 0.220733, 0.246318, 0.249556, 0.251888, 0.253508, 0.254674, 0.257459, 0.388196, 0.434113, 1.074880 and 1.074950. We insist that with the SAT-solver, we find some solutions, but we cannot easily classify them. Thus we are unsure we have indeed found the best one. Nonetheless, it is already of great practical importance to exhibit some solutions.

G.6 Conclusions and Perspectives

Masking is a pro-active countermeasure against side-channel attacks. It implies adequately extra random variables amidst the computation in order to remove dependencies between the leakage of computation and guesses of internal sensitive values by a prospective attacker. Based on a representative first-order leakage model, this article explores the connections between the mask entropy and the best achievable security. If the implementation leaks its data values, then the leakage increases in proportion of the mask entropy reduction. Nonetheless, in practice, the implementation leaks a non-bijective value of its internal variables, such as the sum of their n bits. In this case, we show that the leakage is never null when limiting to a subset of few mask values amongst the 2^n possible. Furthermore, higher-order attacks can defeat this protection even if the

mask losses as little as 1 single bit of entropy. Thus, we explore other mask entropy *vs* security tradeoffs. Our methodology is to demand resistance against CPA and 2O-CPA, and to minimize the leakage.

The criteria for masks selection has been formalized as a condition on the Walsh transform of an indicator function. This criteria has been used heuristically in a SAT-solver, but we expect that constructive methods based on the Boolean theory, for all n, can be invented. We exhibit the best solutions for $n = 4$ and $n = 5$, and prove the existence of varied values of mutual information for some masks cardinality for $n = 8$ (thanks to the SAT-solver). We notably show that amongst the masks subsets that allow for a resistance at orders 1 and 2 against CPA, some are less sensitive to MIA than others, especially for $\mathsf{Card}[\mathcal{M}] = 16$. Therefore, there is a real opportunity for the designer to reduce the cost of the countermeasure in a reasoned way. We insist that, at first sight, it can seem very audacious to mask an eight bit sensitive data with only four bits of mask. But it is indeed possible due to the high non-injectivity of the HW function, that maps 256 values into only 9.

Controlling the overhead in terms of resources is an enabler for masking technologies. Some countermeasures are expensive and our proposed tradeoff definitely shows that it is possible to quantify the security loss when one downgrades a countermeasure. As a perspective, we note that to further save area and speed, instead of storing the sboxes in RAM and selecting them randomly, we could take advantage of the dynamic partial reconfiguration of modern FPGAs to do so [306]. The idea is that even if computed at full throughput, the attacker does not have enough time to collect enough traces with a consistent set of sboxes to succeed an attack. This assumption is the same as those used for the resilience "leakage-proof" countermeasures.

Acknowledgments

The authors thank Manuel San Pedro for insightful discussions about SAT-solvers, and Sébastien Briais for ideas about the constructions of indicator functions. This work has been partly supported by the French National Research Agency (ANR), under grant ANR-09-SEGI-013 (ARPEGE project **SecReSoC**, "**Sec**ured **R**econfigurable **System o**n **C**hip").

G.7 Appendix 1: If $\mathscr{L}$ is not injective, then $\mathsf{I}[\mathscr{L}(Z \oplus M); Z]$ depends on $\mathcal{M}$, where $Z \sim \mathcal{U}(\mathbb{F}_2^n)$ and $M \sim \mathcal{U}(\mathcal{M})$

This property is exemplified in the following case-study, where $n = 2$, $\mathsf{Card}[\mathcal{M}] = 2$ and $\mathscr{L}$ is defined in Tab. G.6. This leakage function is not meant to be realistic: it is simply an example to illustrate how computations unfold. Let us define $Y \doteq Z \oplus M$. Then $Y \sim \mathcal{U}(\mathbb{F}_2^n)$, and the entropy of $\mathscr{L}(Y)$ is equal to $\mathsf{H}[\mathscr{L}(Z \oplus M)] = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} = \frac{3}{2}$ bit.

In the two next subsections G.7.1 and G.7.2, we compute $\mathsf{I}[\mathscr{L}(Z \oplus M); Z]$. We recall

Table G.6: Imaginary truth-table of $\mathscr{L} : \mathbb{F}_2^2 \to \mathbb{R}$, used in subsections G.7.1 and G.7.2.

y	$\mathscr{L}(y)$
00	0
01	0
10	1
11	2

Table G.7: Memento for the computation of conditional probabilities and entropies when $\mathscr{L}$ is defined in Tab. G.6 and $\mathcal{M} = \{00, 01\}$.

z	$\mathsf{P}[\mathscr{L}(z \oplus M) = \ell]$			$\mathsf{H}[\mathscr{L}(z \oplus M)]$
	$\ell = 0$	$\ell = 1$	$\ell = 2$	
00	$\mathsf{P}[00] + \mathsf{P}[01] = \frac{1}{2} + \frac{1}{2} = 1$	$\mathsf{P}[10] = 0$	$\mathsf{P}[11] = 0$	0
01	$\mathsf{P}[01] + \mathsf{P}[00] = \frac{1}{2} + \frac{1}{2} = 1$	$\mathsf{P}[11] = 0$	$\mathsf{P}[10] = 0$	0
10	$\mathsf{P}[10] + \mathsf{P}[11] = 0 + 0 = 0$	$\mathsf{P}[00] = \frac{1}{2}$	$\mathsf{P}[01] = \frac{1}{2}$	1
11	$\mathsf{P}[11] + \mathsf{P}[10] = 0 + 0 = 0$	$\mathsf{P}[01] = \frac{1}{2}$	$\mathsf{P}[00] = \frac{1}{2}$	1

that, for all random variable X and all ℓ belonging to the image of $\mathbb{F}_2^n$ by $\mathscr{L}$:

$$\begin{aligned} \mathsf{P}[\mathscr{L}(Y) = \ell] &= \sum_{y \in \mathbb{F}_2^n} \mathsf{P}[\mathscr{L}(Y) = \ell \mid Y = y] \cdot \mathsf{P}[Y = y] \\ &= \sum_{\substack{y \in \mathbb{F}_2^n \\ \mathscr{L}(y) = \ell}} \mathsf{P}[Y = y] = \sum_{y \in \mathscr{L}^{-1}(\ell)} \mathsf{P}[Y = y]\,. \end{aligned}$$

Also, $\mathsf{H}[\mathscr{L}(Y)] = -\sum_{\ell \in \mathscr{L}(\mathbb{F}_2^n)} \mathsf{P}[\mathscr{L}(Y) = \ell] \log_2 \mathsf{P}[\mathscr{L}(Y) = \ell]$.

G.7.1 $\mathcal{M} = \{00, 01\}$

Some intermediate computations are detailed in Tab. G.7. They allow to derive that the mutual information $\mathsf{I}[\mathscr{L}(Z \oplus M); Z]$ is equal to $\frac{3}{2} - \left(\frac{1}{4} \cdot 0 + \frac{1}{4} \cdot 0 + \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 1\right) = 1$ bit.

G.7.2 $\mathcal{M} = \{01, 10\}$

Some intermediate computations are detailed in Tab. G.8. These results yield that the mutual information $\mathsf{I}[\mathscr{L}(Z \oplus M); Z]$ is equal to $\frac{3}{2} - \left(\frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 1\right) = \frac{1}{2}$ bit. Consequently, this choice of $\mathcal{M}$ is better than the previous one.

Table G.8: Memento for the computation of conditional probabilities and entropies when $\mathscr{L}$ is defined in Tab. G.6 and $\mathcal{M} = \{01, 10\}$.

z	$\mathsf{P}[\mathscr{L}(z \oplus M) = \ell]$			$\mathsf{H}[\mathscr{L}(z \oplus M)]$
	$\ell = 0$	$\ell = 1$	$\ell = 2$	
00	$\mathsf{P}[00] + \mathsf{P}[01] = 0 + \frac{1}{2} = \frac{1}{2}$	$\mathsf{P}[10] = \frac{1}{2}$	$\mathsf{P}[11] = 0$	1
01	$\mathsf{P}[01] + \mathsf{P}[00] = \frac{1}{2} + 0 = \frac{1}{2}$	$\mathsf{P}[11] = 0$	$\mathsf{P}[10] = \frac{1}{2}$	1
10	$\mathsf{P}[10] + \mathsf{P}[11] = \frac{1}{2} + 0 = \frac{1}{2}$	$\mathsf{P}[00] = 0$	$\mathsf{P}[01] = \frac{1}{2}$	1
11	$\mathsf{P}[11] + \mathsf{P}[10] = 0 + \frac{1}{2} = \frac{1}{2}$	$\mathsf{P}[01] = \frac{1}{2}$	$\mathsf{P}[00] = 0$	1

Table G.9: Memento for the computation of conditional probabilities and entropies when $\mathscr{L} = \mathsf{HW}$.

$\mathcal{M} = \{00, 11\}$

z	$\mathsf{P}[\mathscr{L}(z \oplus M) = \ell]$			$\mathsf{H}[\mathscr{L}(z \oplus M)]$
	$\ell = 0$	$\ell = 1$	$\ell = 2$	
00	$\frac{1}{2}$	0	$\frac{1}{2}$	1
01	0	1	0	0
10	0	1	0	0
11	$\frac{1}{2}$	0	$\frac{1}{2}$	1

$\mathcal{M} = \{01, 10\}$

z	$\mathsf{P}[\mathscr{L}(z \oplus M) = \ell]$			$\mathsf{H}[\mathscr{L}(z \oplus M)]$
	$\ell = 0$	$\ell = 1$	$\ell = 2$	
00	0	1	0	0
01	$\frac{1}{2}$	0	$\frac{1}{2}$	1
10	$\frac{1}{2}$	0	$\frac{1}{2}$	1
11	0	1	0	0

G.7.3 Other Case Study

If $\mathscr{L} = \mathsf{HW}$, then $\mathscr{L}(00) = 0$, $\mathscr{L}(01) = \mathscr{L}(10) = 1$ and $\mathscr{L}(11) = 2$. For two $\mathcal{M}$, the conditional probabilities and entropies are given in Tab. G.9.

So, contrarily to the previous $\mathscr{L}$, we now have with $\mathscr{L} = \mathsf{HW}$ that, for both cases, $\mathsf{I}[\mathscr{L}(Z \oplus M); Z] = \frac{3}{2} - \frac{1}{4}(1 + 1) = 1$ bit.

G.8 Appendix 2: Exact Calculation of $\mathsf{H}[\mathsf{HW}(Z)]$ and of $\mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$ when M Takes Two Complementary Values

In general, it is not obvious to compute a mutual information generically. However, for some specific case, it is possible to get an analytical expression. In this section, we compute the "MIA" when $\mathsf{Card}[\mathcal{M}] = 2$, and more precisely $M \in \{m, \neg m\}$.

First of all, the mutual information without countermeasure is equal to: $\mathsf{I}[\mathsf{HW}(Z); Z] = \mathsf{H}[\mathsf{HW}(Z)] = -\sum_{h=0}^{n} \frac{\binom{n}{h}}{2^n} \log_2 \frac{\binom{n}{h}}{2^n}$ bit, because Z is uniformly distributed on $\mathbb{F}_2^n$.

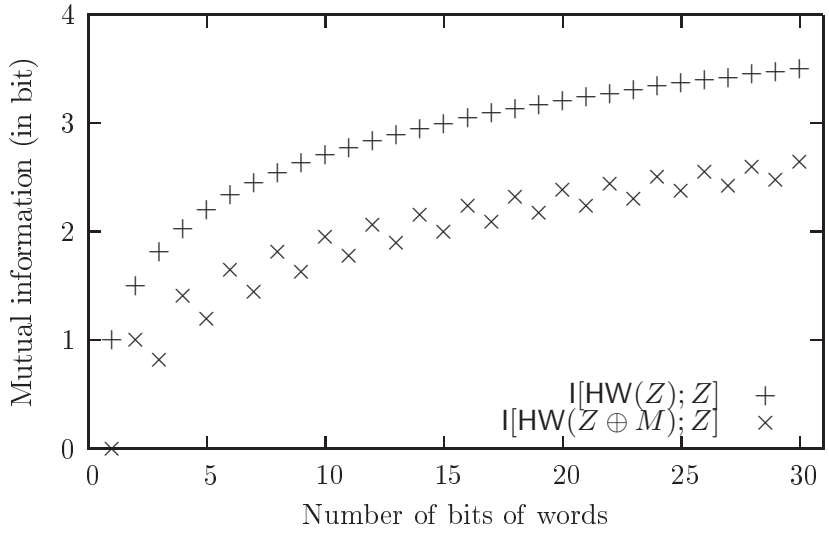


Figure G.2: Mutual information (exploitable by an MIA) without the masking and with masking when M takes two random complementary values with equal probability.

Second, we are interested in the mutual information with the countermeasure, *i.e.* $\mathsf{I}[\mathsf{HW}(Z{\oplus}M); Z] = \mathsf{H}[\mathsf{HW}(Z{\oplus}M)] - \mathsf{H}[\mathsf{HW}(Z{\oplus}M) \mid Z]$. Now, the entropy $\mathsf{H}[\mathsf{HW}(Z{\oplus}M)]$ is equal to $\mathsf{H}[\mathsf{HW}(Z)]$, whatever the distribution of M, because $Z \sim \mathcal{U}(\mathbb{F}_2^n)$. When M takes complementary values, the random variable $(\mathsf{HW}(Z \oplus M) \mid Z = z)$ takes values $\mathsf{HW}(z \oplus m)$ or $\mathsf{HW}(z \oplus \neg m) = n - \mathsf{HW}(z \oplus m)$. Those two values are different, but when $\mathsf{HW}(z \oplus m) = n/2$. When n is odd, this cannot happen. Thus, the random variable $(\mathsf{HW}(Z \oplus M) \mid Z = z)$ takes two equiprobable values, hence has unitary entropy. When n is even, for $\binom{n}{n/2}$ values of z amongst the 2^n possible, the random variable has only one value $n/2$, hence is deterministic. This property is independent on the choice for the mask $m \in \mathbb{F}_2^n$. Therefore, $\mathsf{I}[\mathsf{HW}(Z \oplus M); Z] = \mathsf{I}[\mathsf{HW}(Z); Z] - 1 + \delta(n \mod 2) \times \binom{n}{n/2}/2^n$.

So, to summarize, when masking with two complementary masks, the leaked information in Hamming weight is reduced by:

- exactly one bit if n is odd, but
- less than one bit if n is even. This case is unfavorable, because of an indiscernibility property that make the mask useless in some configurations.

The values of the MIA without and with countermeasure are given in Fig. G.2.

G.9 Appendix 3: Derivation of Eqn. (G.5) and (G.6)

The Eqn. (G.4) for $d = 1$ and 2 is calculated thanks to an alternative form of the variance: $\mathsf{Var}(X) = \mathsf{E}X^2 - (\mathsf{E}X)^2$. Also, in the two following subsections G.9.1 and G.9.2, we abridge "$\sum_{x \in \mathbb{F}_2^n}$", "$\sum_{m \in \mathcal{M}}$" and "$\sum_{i \in [\![1,n]\!]}$" simply by "$\sum_x$", "$\sum_m$", "$\sum_i$", respectively.

G.9.1 Derivation of Eqn. (G.5)

To compute the denominator of Eqn. (G.4), we need to estimate:
- $\mathsf{E}\frac{-1}{2}\sum_i (-1)^{(Z\oplus M)_i} = 0$ (by summation over $z \in \mathbb{F}_2^n$) and
- $\mathsf{E}\left(\frac{-1}{2}\sum_i (-1)^{(Z\oplus M)_i}\right)^2$.

This latter equation writes:

$$\begin{aligned}
& \frac{1}{\mathsf{Card}[\mathcal{M}]}\sum_m \frac{1}{2^n}\sum_z \frac{1}{2^2}\left(\sum_{i=1}^n (-1)^{(z\oplus m)_i}\right)^2 \\
= \; & \frac{1}{2^n\mathsf{Card}[\mathcal{M}]}\sum_m \frac{1}{2^2}\sum_{i_0,i_1}\sum_z (-1)^{(z\oplus m)_{i_0}\oplus(z\oplus m)_{i_1}} \\
= \; & \frac{1}{2^n\mathsf{Card}[\mathcal{M}]}\sum_m \frac{1}{2^2}\sum_{i_0,i_1} 2^n\delta(i_0-i_1) \qquad \begin{array}{l}// \text{ Recall that } \delta \text{ is the} \\ // \text{ Kronecker symbol.}\end{array} \\
= \; & \frac{1}{2^n\mathsf{Card}[\mathcal{M}]}\sum_m \frac{1}{2^2} n 2^n = \frac{n}{4}\,.
\end{aligned} \tag{G.10}$$

Now, the numerator of Eqn. (G.4) involves on the one hand:

$$\mathsf{E}\left(\frac{-1}{2}\sum_i (-1)^{(Z\oplus M)_i}\right)^1 = \frac{-1}{2\cdot 2^n\cdot\mathsf{Card}[\mathcal{M}]}\sum_m\sum_i\left(\sum_z (-1)^{(z\oplus m)_i}\right) = 0\ ,$$

and on the other hand:

$$\begin{aligned}
& \mathsf{E}\left(\mathsf{E}\left(\left(\frac{-1}{2}\sum_i (-1)^{(Z\oplus M)_i}\right)^{d=1} \mid Z\right)\right)^2 \\
= \; & \frac{1}{2^2 2^n\mathsf{Card}[\mathcal{M}]^2}\sum_{m_0,m_1}\sum_{i_0,i_1}\sum_z (-1)^{(z\oplus m_0)_{i_0}\oplus(z\oplus m_1)_{i_1}} \\
= \; & \frac{1}{2^{n+2}\mathsf{Card}[\mathcal{M}]^2}\sum_{m_0,m_1}\sum_i 2^n(-1)^{(m_0\oplus m_1)_i} \\
= \; & \frac{1}{2^2}\sum_i\left(\frac{1}{\mathsf{Card}[\mathcal{M}]}\sum_m (-1)^{m_i}\right)^2 .
\end{aligned}$$

Consequently, we obtain the expression announced in Eqn. (G.5):

$$\rho^{(1)}_{\text{opt}} = \frac{1}{n}\sum_{i=1}^n\left(\frac{1}{\mathsf{Card}[\mathcal{M}]}\sum_{m\in\mathcal{M}}(-1)^{m_i}\right)^2 .$$

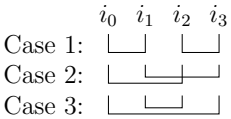


Figure G.3: Pairwise equality relationships in a set of four indices.

G.9.2 Derivation of Eqn. (G.6)

The denominator of Eqn. (G.4), in the case $d = 2$, requires the computation of $\mathsf{E}\left(\frac{-1}{2}\sum_i (-1)^{(Z\oplus M)_i}\right)^2$. It has already been computed in the previous section G.4.1 in Eqn. (G.10). Its value is $n/4$. The second value required for the denominator is: $\frac{1}{2^n \mathsf{Card}[\mathcal{M}]}\sum_m \frac{1}{2^4}\sum_z \left(\sum_i (-1)^{(z\oplus m)_i}\right)^4$. This expression is proportional to:

$$\begin{aligned} & \sum_m \sum_z \left(\sum_{i=1}^{n} (-1)^{(z\oplus m)_i}\right)^4 \\ = & \sum_m \sum_{\substack{i_0,i_1,\\ i_2,i_3}} \sum_z (-1)^{(z\oplus m)_{i_0}\oplus(z\oplus m)_{i_1}\oplus(z\oplus m)_{i_2}\oplus(z\oplus m)_{i_3}} . \end{aligned} \tag{G.11}$$

If $(z\oplus m)_{i_0}\oplus(z\oplus m)_{i_1}\oplus(z\oplus m)_{i_2}\oplus(z\oplus m)_{i_3}$ depends on z, then the summation over z yields zero in Eqn. (G.11). The cases where this expression depends on z are enumerated below:

a) i_0, i_1, i_2 and i_3 are different: it depends on z;

b) Two indices are equal, and the other are different: it depends on z;

c) Two indices are equal and the other two are also equal: it does not depend on z;

d) Three indices are equal and the last one is different: it depends on z;

e) The four indices are equal: it does not depend on z;

Thus, we need to enumerate the cases where indices are equal two by two (which include the last case of equality between all the masks). The possibilities are shown in Fig. G.3, where the identical indices are linked together. Each case happens $n \times (n-1)$ times: n times to choose the first couple, and $n-1$ remaining possibilities for the second couple. We must add n cases for configurations where $i_0 = i_1 = i_2 = i_3$. Thus, the total number of possibilities is $3n(n-1)+n = n(3n-2)$. Therefore, we deduce that $\frac{1}{2^n \mathsf{Card}[\mathcal{M}]}\sum_m \frac{1}{2^4}\sum_z \left(\sum_{i=1}^{n} (-1)^{(z\oplus m)_i}\right)^4 = \frac{n(3n-2)}{4}$.

Eventually, the denominator is equal to: $\frac{1}{4^2}\left(n(3n-2)-n^2\right) = \frac{n(n-1)}{2^3}$.

To compute the numerator, it can be first noted that:

$$\mathsf{E}\left(\mathsf{E}\left(\left(\frac{-1}{2}\sum_i (-1)^{(Z\oplus M)_i}\right)^{d=2}\right) \mid Z\right) = \frac{n}{4} ,$$

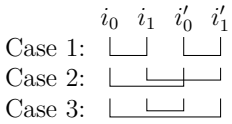


Figure G.4: Pairwise equality relationships in a set of four indices, already belonging to two classes (nominal and primed).

as already demonstrated in Eqn. (G.10). Then, we compute:

$$\begin{aligned}
&\frac{1}{2^n}\sum_z\left(\frac{1}{\mathsf{Card}[\mathcal{M}]}\sum_m\left(\frac{-1}{2}\sum_i(-1)^{(z\oplus m)_i}\right)^{d=2}\right)^2\\
=\;&\frac{1}{2^4 2^n \mathsf{Card}[\mathcal{M}]^2}\sum_z\left(\sum_m\sum_{i_0,i_1}(-1)^{(z\oplus m)_{i_0}\oplus(z\oplus m)_{i_1}}\right)^2\\
=\;&\frac{1}{2^{n+2}\mathsf{Card}[\mathcal{M}]^2}\sum_{m,m'}\sum_{\substack{i_0,i_1,\\ i'_0,i'_1}}\sum_z(-1)^{(z\oplus m)_{i_0}\oplus(z\oplus m)_{i_1}\oplus(z\oplus m')_{i'_0}\oplus(z\oplus m')_{i'_1}}. \qquad \text{(G.12)}
\end{aligned}$$

This summation resembles that depicted in Fig. G.3, but now the indices already refer to some classes: i_0 and i_1 relate to mask m, whereas i'_0 and i'_1 relate to mask m'. This setup is shown in Fig. G.4. In this section, we count the case $i_0 = i_1 = i'_0 = i'_1$ in each case, and we eventually subtract those multiply counted. Therefore:

- In case 1: the masks m and m' cancel out one each other, so the sum is equal to $\frac{1}{2^4}n^2$.
- In case 2: the sum is equal to:

$$\begin{aligned}
&\frac{1}{2^4\mathsf{Card}[\mathcal{M}]^2}\sum_{m,m'}\sum_{\substack{i_0,i_1,\\ i'_0,i'_1}}(-1)^{m_{i_0}\oplus m_{i_1}\oplus m'_{i'_0}\oplus m'_{i'_1}}\times\delta(i_0-i'_0)\times\delta(i_1-i'_1)\\
=\;&\frac{1}{2^4\mathsf{Card}[\mathcal{M}]^2}\sum_{m,m'}\sum_{i_0,i_1}(-1)^{(m\oplus m')_{i_0}\oplus(m\oplus m')_{i_1}}\\
=\;&\frac{1}{2^4\mathsf{Card}[\mathcal{M}]^2}\sum_{m,m'}\left(\sum_i(-1)^{(m\oplus m')_i}\right)^2.
\end{aligned}$$

- In case 3: it yields the same as case 2, because of the invariance of Eqn. (G.12) in $i_0 \leftrightarrow i_1$ and in $i'_0 \leftrightarrow i'_1$.

Now, the equality between the four indices is counted three times, and thus shall be subtracted twice. Therefore, the numerator for Eqn. (G.4) when $d = 2$ is equal to:

$$\begin{aligned} & \frac{1}{2^4}\left(\cancel{n^2} + 2 \times \frac{1}{2^4 \mathsf{Card}[\mathcal{M}]^2} \sum_{m,m'} \left(\sum_i (-1)^{(m \oplus m')_i}\right)^2 - 2n\right) - \cancel{(n-1)^2} \\ = \; & \frac{1}{2^3}\left(\frac{1}{\mathsf{Card}[\mathcal{M}]^2} \sum_{m,m'} \left(\sum_i (-1)^{(m \oplus m')_i}\right)^2 - n\right). \end{aligned}$$

The optimal correlation coefficient for a second-order attack is thus equal to the expression already disclosed in Eqn. (G.6):

$$\rho^{(2)}_{\text{opt}} = \frac{1}{n(n-1)}\left(\frac{1}{\mathsf{Card}[\mathcal{M}]^2} \sum_{m,m'} \left(\sum_i (-1)^{(m \oplus m')_i}\right)^2 - n\right). \tag{G.13}$$

G.10 Appendix 4: More Details About the Solutions for $n = 5$ and $n = 8$

G.10.1 All the Solutions that Cancel $\rho^{(1,2)}_{\text{opt}}$ for $n = 5$

The 1057 solutions for $n = 5$ can be grouped by equivalence classes, that consist in permuting the bits or complementing them (when $\mathsf{Card}[\mathcal{M}] = 2^{n-1}$). The table G.10 complements Tab. G.4 by giving in addition the smallest element that generates each class. Also, it can be noticed that if a class of functions of Hamming weight $h \neq 2^n$ exists, then a class of functions of Hamming weight $2^n - h$ also exists. This is due to the complementary identity discussed in Sec. G.4.4.

G.10.2 Detail of the the First Solutions Given in Tab. G.5 for $n = 8$

The lowest possible number of mask values to achieve CPA and 2O-CPA resistance for $n = 8$ is $\mathsf{Card}[\mathcal{M}] = 12$. Many different classes are found, but they share the same metrics, *i.e.* those indicated in Tab. G.5. As an example, with 57 seed values, 14 non-equivalent solutions are found. They are listed below:

0x0200000000400800000400200000100000004001002000008000000001000008,
0x1000080000000001000800000040200000000040800200000100002000000400,
0x0000002080000001004008000100000002040000000040000001000002000008,
0x0000000440200000010020000000000800028000000001000040000008000010,
0x0000080020000004010000200040000000810000000010000000400008000002,
0x0001800000000100002000000800004000000008204000000400100000000002,
0x2000040000000004000000002800100000010000000080200040000800000100 0,
0x2000000000080400000180000000001000400100000000020000000842000000,
0x0004002000004000010000000080020000000800100200008000001000000004,

Table G.10: Complete list of generators of Boolean functions $f : \mathbb{F}_2^5 \to \mathbb{F}_2$ that cancel $\rho_{\mathrm{opt}}^{(1,2)}$.

$\mathsf{HW}(f)$	$\mathsf{I}[\mathsf{HW}(Z \oplus M); Z]$	$\mathsf{I}[Z \oplus M; Z]$	$d_{\mathrm{alg}}^{\circ}(f)$	Generators of classes
8	0.32319	2	2	{ 0x06609009, 0x06909006, 0x81182442 }
12	0.18595	1.41504	3	{ 0x1698a443, 0x19a4c216, 0x83586429, 0x83589426 }
16	0.08973	1	1	{ 0x0ff0f00f, 0x96969696 }
16	0.08973	1	2	{ 0x1bd8e427, 0x87b4d21e }
16	0.12864	1	2	{ 0x1be4e41b, 0x2dd2e11e, 0x8778b44b, 0x96969966 }
16	0.16755	1	1	{ 0x3cc3c33c, 0x96699669 }
16	0.26855	1	2	{ 0x17e8e817, 0x8778e11e, 0x2bd4d42b, 0x99969666 }
16	0.32495	1	2	{ 0x1ee1e11e, 0x2dd2d22d, 0x69969696, 0x87787887, 0x96699696, 0x96969669 }
16	1	1	1	{ 0x69969669 }
20	0.07349	0.67807	3	{ 0x3ddae697, 0x6bd9e53e, 0x97bcda67, 0x9bd6e53e }
24	0.04300	0.41504	2	{ 0x6ff6f99f, 0x9ff6f69f, 0xbddbe77e }
32	0	0	0	{ 0xffffffff }

0x0000040000800010100800000000020000180000200000000000002040000004,
0x8000000002004000000001801000000000102000000004040000000000082000,
0x0001200004000000000000080020400000000040008002001800000000000001,
0x0002000010000400800000400000000200042000000000800000010008100000,
0x0000004042000000000801000000002000012000000000088000000000100400.

The algebraic normal form of the first solution writes as follows: $f(x) = x_2x_1 \oplus x_3x_2x_1 \oplus x_4x_2x_1 \oplus x_4x_3x_2x_1 \oplus x_5x_2x_1 \oplus x_5x_3x_2x_1 \oplus x_5x_4 \oplus x_5x_4x_1 \oplus x_5x_4x_2 \oplus x_5x_4x_3 \oplus x_5x_4x_3x_1 \oplus x_5x_4x_3x_2 \oplus x_6x_2x_1 \oplus x_6x_3x_2x_1 \oplus x_6x_4x_2x_1 \oplus x_6x_4x_3x_2x_1 \oplus x_6x_5x_2x_1 \oplus x_6x_5x_3x_2x_1 \oplus x_6x_5x_4 \oplus x_6x_5x_4x_1 \oplus x_6x_5x_4x_2 \oplus x_6x_5x_4x_3 \oplus x_6x_5x_4x_3x_1 \oplus x_6x_5x_4x_3x_2 \oplus \mathbf{x_6x_5x_4x_3x_2x_1} \oplus x_7x_2x_1 \oplus x_7x_3x_2x_1 \oplus x_7x_4x_2x_1 \oplus x_7x_4x_3x_2x_1 \oplus x_7x_5x_2x_1 \oplus x_7x_5x_3x_1 \oplus x_7x_5x_4 \oplus x_7x_5x_4x_1 \oplus x_7x_5x_4x_2 \oplus x_7x_5x_4x_3 \oplus x_7x_5x_4x_3x_2 \oplus \mathbf{x_7x_5x_4x_3x_2x_1} \oplus x_7x_6 \oplus x_7x_6x_1 \oplus x_7x_6x_2 \oplus x_7x_6x_3 \oplus x_7x_6x_3x_1 \oplus x_7x_6x_3x_2 \oplus x_7x_6x_4 \oplus x_7x_6x_4x_1 \oplus x_7x_6x_4x_2 \oplus x_7x_6x_4x_3 \oplus x_7x_6x_4x_3x_1 \oplus \mathbf{x_7x_6x_4x_3x_2x_1} \oplus x_7x_6x_5 \oplus x_7x_6x_5x_1 \oplus x_7x_6x_5x_2 \oplus x_7x_6x_5x_3 \oplus x_7x_6x_5x_3x_2 \oplus \mathbf{x_7x_6x_5x_3x_2x_1} \oplus \mathbf{x_7x_6x_5x_4x_2x_1} \oplus \mathbf{x_7x_6x_5x_4x_3x_1} \oplus \mathbf{x_7x_6x_5x_4x_3x_2} \oplus x_8x_2x_1 \oplus x_8x_3x_2x_1 \oplus x_8x_4x_2x_1 \oplus x_8x_4x_3 \oplus x_8x_4x_3x_1 \oplus x_8x_4x_3x_2 \oplus x_8x_5x_2x_1 \oplus x_8x_5x_3x_2x_1 \oplus x_8x_5x_4 \oplus x_8x_5x_4x_1 \oplus x_8x_5x_4x_2 \oplus \mathbf{x_8x_5x_4x_3x_2x_1} \oplus x_8x_6x_2x_1 \oplus x_8x_6x_3x_1 \oplus x_8x_6x_4x_2x_1 \oplus x_8x_6x_4x_3 \oplus x_8x_6x_4x_3x_2 \oplus \mathbf{x_8x_6x_4x_3x_2x_1} \oplus x_8x_6x_5x_2 \oplus x_8x_6x_5x_3x_1 \oplus x_8x_6x_5x_3x_2 \oplus \mathbf{x_8x_6x_5x_3x_2x_1} \oplus x_8x_6x_5x_4 \oplus x_8x_6x_5x_4x_1 \oplus \mathbf{x_8x_6x_5x_4x_2x_1} \oplus \mathbf{x_8x_6x_5x_4x_3x_1} \oplus \mathbf{x_8x_6x_5x_4x_3x_2} \oplus x_8x_7x_2x_1 \oplus x_8x_7x_3x_2x_1 \oplus x_8x_7x_4x_3 \oplus x_8x_7x_4x_3x_1 \oplus x_8x_7x_4x_3x_2 \oplus \mathbf{x_8x_7x_4x_3x_2x_1} \oplus x_8x_7x_5x_2x_1 \oplus x_8x_7x_5x_3x_1 \oplus x_8x_7x_5x_3x_2 \oplus \mathbf{x_8x_7x_5x_3x_2x_1} \oplus x_8x_7x_5x_4 \oplus x_8x_7x_5x_4x_1 \oplus x_8x_7x_5x_4x_2 \oplus \mathbf{x_8x_7x_5x_4x_2x_1} \oplus \mathbf{x_8x_7x_5x_4x_3x_1} \oplus \mathbf{x_8x_7x_5x_4x_3x_2} \oplus x_8x_7x_6 \oplus x_8x_7x_6x_1 \oplus x_8x_7x_6x_2 \oplus x_8x_7x_6x_3 \oplus x_8x_7x_6x_3x_2 \oplus \mathbf{x_8x_7x_6x_3x_2x_1} \oplus x_8x_7x_6x_4 \oplus x_8x_7x_6x_4x_1 \oplus x_8x_7x_6x_4x_2 \oplus \mathbf{x_8x_7x_6x_4x_2x_1} \oplus \mathbf{x_8x_7x_6x_4x_3x_1} \oplus \mathbf{x_8x_7x_6x_4x_3x_2} \oplus x_8x_7x_6x_5 \oplus x_8x_7x_6x_5x_1 \oplus \mathbf{x_8x_7x_6x_5x_2x_1} \oplus x_8x_7x_6x_5x_3 \oplus \mathbf{x_8x_7x_6x_5x_3x_1} \oplus \mathbf{x_8x_7x_6x_5x_3x_2} \oplus \mathbf{x_8x_7x_6x_5x_4x_1} \oplus \mathbf{x_8x_7x_6x_5x_4x_2} \oplus \mathbf{x_8x_7x_6x_5x_4x_3}$.

In this expression, the products of 6 variables are shown in bold font. It is thus clear that the algebraic degree of f is $d^{\circ}_{\mathrm{alg}}(f) = 6$. The corresponding twelve masks are:

{ 0x03, 0x18, 0x3f, 0x55, 0x60, 0x6e, 0x8c, 0xa5, 0xb2, 0xcb, 0xd6, 0xf9 }.

For the next masks subsets ($\mathsf{Card}[\mathcal{M}] = 16$), there are also different classes. But their MIA do differ, as represented in Fig. G.5 for elements of 286 different classes. Most solutions have algebraic degree 5, but some have 4. The mutual information for algebraic degree 5 is more spread than for $d^{\circ}_{\mathrm{alg}}(f) = 4$. The best solution found by the SAT-solver has an MIA of 0.181675 bit.

Eventually, we mention that for $\mathsf{Card}[\mathcal{M}] > 16$, there still exists different solutions when $\mathsf{Card}[\mathcal{M}] \in \{12 + 4\kappa, 0 \leqslant \kappa \leqslant 61\}$, but that the MIA are less spread. For instance, for $\mathsf{Card}[\mathcal{M}] = 20$, we have found these MIA values: 0.191514, 0.197768, 0.200909, 0.201735, 0.201907, 0.202508, 0.215823, 0.219964, 0.220303, 0.221462, 0.223525, 0.224186, 0.224328, 0.224450, 0.224958 and 0.228925.

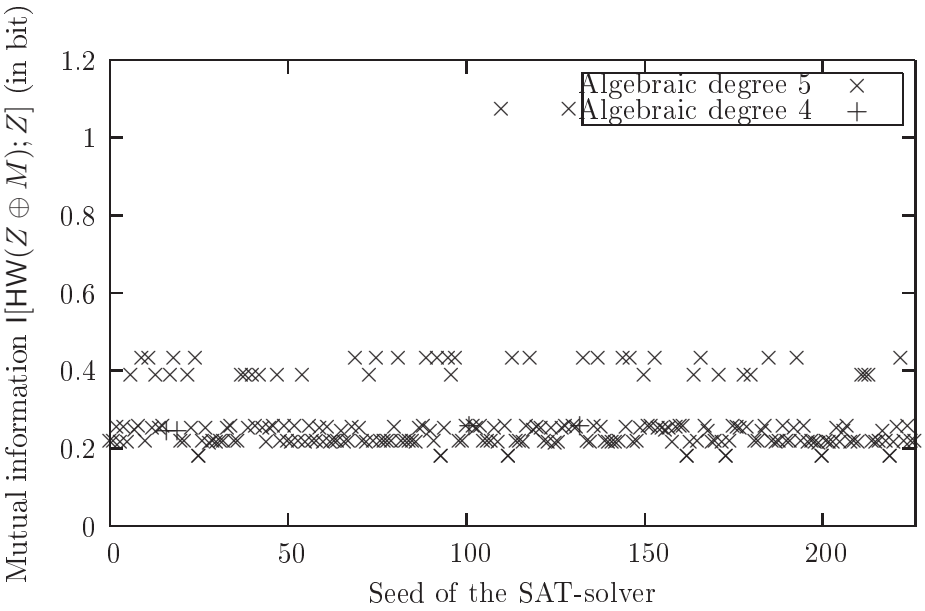


Figure G.5: Mutual information of the leakage in Hamming weight with the $n = 8$-bit sensitive variable Z, for many nonequivalent solutions f of weight $\hat{f}(0) = 16$ that cancels $\rho_{\text{opt}}^{(1,2)}$ found by the SAT-solver.

Appendix H

Combined Side-Channel Attacks

Extended version of article [3]

Authors: *Abdelaziz M. Elaabid, Olivier Meynard, Sylvain Guilley and Jean-Luc Danger*

Abstract

The literature about side-channel attacks is very rich. Many side-channel distinguishers have been devised and studied; in the meantime, many different side-channels have been identified. Also, it has been underlined that the various samples garnered during the same acquisition can carry complementary information. In this context, there is an opportunity to study how to best combine many attacks with many leakages from different sources or using different samples from a single source. This problematic has been evoked as an open issue in recent articles. In this paper, we bring two concrete answers to the attacks combination problem. First of all, we experimentally show that two partitionings can be constructively combined. Then, we explore the richness of electromagnetic curves to combine several timing samples in such a way a sample-adaptative model attack yields better key recovery success rates than a mono-model attack using only a combination of samples (via a principal component analysis). We also extend the list of open problems in the context of attack combinations.

Key words: Side-channel analysis; leakage models; attacks combination; multi-partitioning attacks; multi-modal leakage.

H.1 Introduction

Trusted computing platforms resort to secure components to conceal and manipulate sensitive data. Such components are in charge of implementing cryptographic protocols; for instance, the component is typically asked to encrypt the data with a cryptographic key. The secret key is protected against a direct readout from the circuit thanks to

tamper-proof techniques. In general, the component is shielded by coatings to protect it from malevolent manipulations (active or passive micro-probing [133], modification, *etc.*). However, it has been noted that despite this protection, some externally measurable quantities can be exploited without touching the component. Typically, without special care, internal data are somehow modulating the computation timing, the instant current drawn from the power supply, and the radiated fields. Thus, those unintentional physical emanations can be analyzed in a view to derive from them some sensitive information. Such analyses are referred to as side-channel attacks. The way the observed measurements are affected by the internal data is *a priori* unknown by the attacker, although in some cases an hypothetical, hence imperfect, physical model can be assumed. The link between the data and the side-channel is called the leakage model.

Most side-channel attacks start by a tentative partitioning of the measurements, indexed by key hypotheses [433]. Then, the adversary assesses the quality of each partitioning. This information is typically summarized by a figure of merit. This figure of merit can be a difference of means (in case there are only two partitions [248]), a correlation (case of the CPA [60]), a likelihood (case of template attacks [69]) or a mutual information (case of the MIA [141]), to cite only the few most widespread. Such figures of merit are often referred to as distinguishers, as they are able to successfully distinguish between the key candidates to select the correct one. The comparison of these distinguishers on the same acquisitions has been already discussed in some papers [89, 290, 257, 142, 468]. It appears that for a given partitioning, some distinguishers are better than the others to rank the correct key first, some other distinguishers are better than the others to optimize the average rank of the correct key [142]. Moreover, the conclusions depend on the target, since the leakage structure is inherent to each device. The definition of new distinguishers is an active research area; indeed, every new distinguisher contributes to feed a battery of attacks suitable to be launched in parallel on a device under test.

Another research direction is to attempt to make the most of the existing distinguishers. One interesting option is to constructively combine the wealth of cited attacks on common side-leakage traces. Another option consists in combining different samples of traces or even different traces acquired concomitantly. All in one, various types of combinations can be envisioned. To our best knowledge, most of those suggestions are nearly virgin problems:

1. Various distinguishers for a same partitioning can be combined;
2. One distinguisher can be evaluated on various partitionings;
3. The diversity can also come from the multiplicity of timing samples usually garnered during an acquisition campaign;
4. It can also arise from multi-modal acquisitions;
5. There can be situations where the most suitable partitioning can evolve from sample to sample in a side-channel capture.

The first point is still open; some attempts have made in the first edition of the DPA contest [445] (contribution of Jung HAE-IL from Korea University); however, it remains unclear how different distinguishers can reinforce mutually from a common set of data.

Regarding the second point, it has already been observed that even an unprotected device might leak differently for different partitionings [118]. In this article, it is proved that there are circuits for which those leakages are statistically independent; therefore, it is profitable to combine them, since the result of the attack will certainly be improved by using multiple partitionings simultaneously.

The third example has already been discussed once in the literature: the so-called MMIA (multi-valued MIA [140]) exploits two different samples from leakage traces in a view to defeat a masking countermeasure. The idea is that the joint distribution of those two samples depends on the secret, and that the joint likelihood or mutual information is therefore a distinguisher.

Alternatively, as listed in item four, a similar setup can also consist of several side-channels. The multi-channel paper [5] explain how to best select channels to be combined. For instance, the channels can be the result of the demodulation of same EM wave for several carrier frequencies. The acquisitions can also be conducted in parallel according to different sensors (such as power and electromagnetic field, as suggested in [432]). Alternatively, the multiple channels can be two identical sensors but recording the emanations from two different locations over the chip, which is feasible on large-scale circuits, such as FPGAs. The best localization of the two sensors can be investigated by initially performing a cartography [375] of the device.

A preliminary study of T.-H. Le in Chapter 4 of her PhD thesis [255] suggests that there is little benefit to gain from multiple acquisitions using the same modality, namely the magnetic field, acquired from different locations. The same conclusion is drawn in [259] by the use of independent component analysis (ICA), which leads to poorly conditioned matrices, hence numerically unsolvable equation systems. Nonetheless, this case study targeted a smartcard, that is quasi-punctual with respect to the wavelengths of interest [1]. Larger circuits, such as FPGAs or ASICs on complex PCBs, could revive the interest in such constructive interference methods.

Eventually, the fifth problem is somehow related the third one: the leakage model depends on the temporal samples. However, we target in this topic situations where the difference of nature is not artificially due to a countermeasure, but naturally by the distortion into the communication channel between the leaking device and the side-channel sensor. This behavior is expected for instance to exist in magnetic waves produced by a PCB. We illustrate this situation on a real example where the leakage does evolve during the encryption. As in the case of the second issue, we emphasize that an initial search of samples of interest using general methods would have led us to neglect the richness of this behavior. This makes this problem all the more interesting from the characterization point of view. Typically, this opens the door to a technique to search independent information partitions in time, such as computing the traces' internal mutual information.

To be completely exhaustive, we mention that other kinds of combinations have already been studied. For instance, those papers [10, 87] describe the combination of

1. Indeed, the wavelengths of interest are greater than $c/f_{\max} \approx 5$ mm for an acquisition chain of bandwidth $[0, f_{\max} = 3]$ GHz.

two attack paths, namely passive (observation) and active (perturbation) analyses. Such attack strategies are out of the scope of this article.

The rest of the paper is structured as follows. The section H.2 tackles the question of the multiple-partitioning attacks. The section H.3 reports an original multi-sample electromagnetic (EM) trace, where the leakage model depends on the sample within the trace. We investigate attacks that could take advantage of this originally rich leakage and show that a combined attack indeed outperforms classical ones. The conclusions and the perspectives are in Sec. H.4.

H.2 Combined Attacks and Metrics based on Multiple Partitions

We explore in this section the combination of multiple partitionings on template attacks. Indeed, some "comparison" attacks that require a physical model of the leakage fail if the leakage function does not match enough the leaking modality of the device.

In [434], a framework is presented in order to evaluate the security of a cryptographic device. This approach relies on two different views: on the one hand the robustness of a circuit against a leakage function, and on the other the strength of an adversary. The information theory and specially the conditional entropy is chosen to quantify the information leaked during encryption. This very concept is thus promoted in order to measure the robustness. Indeed, the more the circuit is leaking the more it is vulnerable. The strength of the adversary is determined for example by its success rate to retrieve the encryption key.

H.2.1 Information Theoretic Metric

We adopt the idea that the quality of a circuit is assessed by the amount of information given by a leakage function. Thus, if S_K is the random variable representing the secret (ideally the key values), and $\mathbf{L}$ is the random variable representing the values of the leakage function.
The residual uncertainty on S_K knowing $\mathbf{L}$ is given by $\mathbf{H}(S_K \mid L)$. $\mathbf{H}$ is the conditional entropy introduced by Claude E. Shannon [434, 118]. Note that this value will depend on sensitive variables chosen, and thus the quality of the leakage function. The more the sensitive variable leaks, the smaller is the entropy and more vulnerable is the circuit.

H.2.2 Template Attacks

Template attacks are among the most powerful forms of side channel attacks. They are able to break implementations and countermeasures which assumes that the attacker cannot get more than a very small number of samples extracted from the attacked device. To this end, the adversary needs a hardware identical to the target, which allows him to obtain some information under the form of leakage realizations. The main step is to perform a modeling process; its goal is to build classes for side-channel traces that will

help identify the secret values during the on-line phase of the attack. Said differently, the information provided by profiling are used to classify some part of encryption key. Actually, the full round key has obviously too many bits to be guessed in one go by exhaustive search. In general, the key bits at entering substitution boxes (sboxes) are targeted. In fact, they all contribute to activate the same logic, which explains why it is beneficial to guess them together. An adversary can also select other key bits if they are more vulnerable. In other words, the attacker itself selects the bits of the key best for his attack. Guessing the correct key is a problem of decision theory. To solve it, we introduce a statistical model that is directly applicable in principle to the problem of classification. This application is mainly based on Bayes' rule, which allows to evaluate an *a posteriori* probability (that is after the effective observation), knowing the conditional probability distributions *a priori* (*i.e.* independent of any constraint on observed variables). The maximum likelihood approach helps provide the most appropriate model.

H.2.2.1 Profiling Process.

For this step, we need a set of traces $\mathcal{S}_o, o \in [0, N'[$ corresponding to each N' operation that are also values of the sensitive variable. Traces, denoted by t, are vectors of N dimensions related to random values of plaintext and keys needed to algorithm encryption. These observations are then classified according to functions of leakage $\mathcal{L}$. These leakage functions must depend on the configuration of the circuit, and of the implemented algorithm. This provides a framework for the estimation of the leakage during encryption. For each set $\mathcal{S}_o, o \in [0, N'[$ the attacker computes the average $\mu_o = \frac{1}{|\mathcal{S}_o|} \sum_{t \in \mathcal{S}_o} t$ and the covariance matrix $\Sigma_o = \frac{1}{|\mathcal{S}_o|-1} \sum_{t \in \mathcal{S}_o} (t - \mu_o)(t - \mu_o)^\mathsf{T}$. The ordered pair (μ_o, Σ_o) associated with value o of the leakage function outputs, is called *template* and will be used in the attack to retrieve subkeys. It allows to build the ideal probability density function (PDF) of a multivariate Gaussian distribution.

H.2.2.2 Principal Component(s) Analysis.

One of the main contributions of the template attack is that an adversary may use all the information given by any trace. However, he is confronted with enormous data he has on hand, especially the covariance matrices. This poses some difficulties for calculations, since, because of algorithmic noise, large covariance matrices are poorly conditioned. For this purpose, the principal component analysis (PCA) is used to get round those drawbacks. It allows to analyze the structure of the covariance matrix (variability, dispersion of data). The aim of PCA is to reduce the data to $q \ll N$ new descriptors, that summarize a large part of (if not all) the variability. Also, it allows to better visualize the data in 2 or 3 dimensions (if $q = 2$ or 3).

These new descriptors are given by the data projection on the most significant eigenvectors given by PCA. Let EV be the matrix containing the eigenvectors classified according to the decreasing eigenvalues. The mean traces and covariance matrices are then expressed in this basis by: $p\mu_o = (EV)^\mathsf{T} \mu_o$ and $P\Sigma_o = (EV)^\mathsf{T} \Sigma_o (EV)$.

H.2.2.3 Online Attack and Success Rate.

The *online attack* consists in first capturing one trace t of the target device during an encryption using the secret key κ. Knowing that each trace corresponds to one leakage value, the secret key will be retrieved from this trace by using maximum likelihood: $\kappa = \text{argmax}_{s_{Kc}} Pr(s_{Kc} \mid t)$, where s_{Kc} is the candidate key. Indeed, for each key candidate, we estimate the value of leakage by using the message or the ciphertext that are *a priori* known. The success rate is given by the average number of times where the adversary succeeds to retrieve the key $s_{Kc} = \kappa$. For each attempt the adversary can use one trace corresponding to one query, or a set of traces corresponding to different queries.

H.2.3 Sensitive Variables

In the paper [118] a study is made on the choice of the best suited sensitive variable for an adversary attacking publicly available traces [445]. From a comparison between five different models, it is shown that the most appropriate model for the targeted circuit is the Hamming distance between two registers. However, "partitioning attacks" (in the sense of [433]) on various sensitive values (such as the linear and nonlinear functions inputs) also allows an adversary to recover the key, but with many more traces. The knowledge of circuit architecture provides definitely much more information about the main leakage function. In this article we elaborate by combining these models to retrieve the key with fewer traces, and watch the behavior of entropy as a function of the number of eigenvectors retained in the attack.

H.2.3.1 Combined Models.

The goal is to combine two partitionings. The security of the resulting compound model is evaluated by template attacks; identically, the robustness of the circuit is measured under this new model. Can an adversary that combines models be considered as "higher order" [307]? Is he able to recover the secret key faster? The experiment described in this section attempts to address these issues. Let

1. **Model M1** be the value of the first round corresponding to the fanout of the first sbox. It is a 4-bit model, and
2. **Model M2** be the first bit transition of model M1. It is a mono-bit model, belonging to the general class of "Hamming distance" models.

From those two models, we derive a third one referred to as **Model M3**. M3 combines the 4-bit model M1 and the 1-bit model M2. In other words, M3 is considered as a "bit-field structure" where the value of the most significant bit (MSB) is the model M2. The others 4 bits correspond to the model M1. M3 is the concatenation of MA and M2, and we note M3 $\doteq$ (M1, M2). Hence M3 is a $4 + 1 = 5$ bit model, which means that M3 is based on 32 partitions. Said differently, the partitioning for M3 is equal to the Cartesian product of that of M1 and M2.

The fair comparison between the models is not a trivial operation. Typically, the number of templates for models M1, M2 and M3 differs. Basically, regarding the training (*i.e.* templates building) phase:

1. either the adversary has an equal number of traces by classes.
2. or the adversary has an equal number of traces for all the set of classes.

The choice will influence the success rate as we will see in the forthcoming experiment. The first case is the most realistic: it consists in saying that the precharacterization time is almost unbounded; the valuable asset being the traces taken on-line from the attacked device. We model this situation by taking the same number of traces for each partition. Therefore, in total, much less training traces are used for mono-partition models; but this really represents the case where models are evaluated with as identical conditions as possible. The second one reflects the case where the precharacterization cost is non-negligible. Under this assumption, the advantage of combined attacks is less clear, since the number of available traces to estimate each template gets lower. Thus, in a single-model attack, the greater accuracy of the templates will certainly compensate the loss of benefit conveyed by the combination.

H.2.3.2 First Choice: Matching-Limited Evaluation.

We use an equal number of traces per class. In our experiment we take 1,000 traces per class for models **M1**, **M2**, and **M3**. The comparison is made with and without the use of the thresholding method as presented in [118]. This method consists in accelerating the estimation of the principal directions in a PCA by forcing to zero the samples that are too small in the eigenvectors. The Fig. H.1 illustrates the method. The idea is that most samples with low amplitude would actually be equal to zero with more traces in the estimation of the PCA. The thresholding allows to filter those samples out, so that they do not bring noise to the protection. In the same time, the thresholding keeps the samples with the greatest variance, which makes it a good tool to separate POIs from others. There is of course a trade-off in the choice for the best threshold. A too small threshold keeps too many irrelevant samples, whereas a too large threshold filters out even some weak POIs. For the implementation studied in this section, we found that a value of 40 % is a fair compromise. The figure H.2 shows the success rate of the template attacks with the three models. We recall that the higher the success rate, the better the attack. We see in Fig. H.2 that in the case of non-thresholding, the template attack based on the combined model is better than that on other models. It is much better than model **M1**, and slightly better than model **M2**.

Incidentally, when we resort to thresholding, the model M2 and M3 are equivalent and obviously always better than M1, that models in a less appropriate way the leakage function. The fact only the first PCA eigenvector is used in the comparison accounts for the equivalence between M2 and M3. Indeed, the other eigenvectors among the 31 possible in the case of combined model M3 also contain information, while the model M2 has only one significant direction.

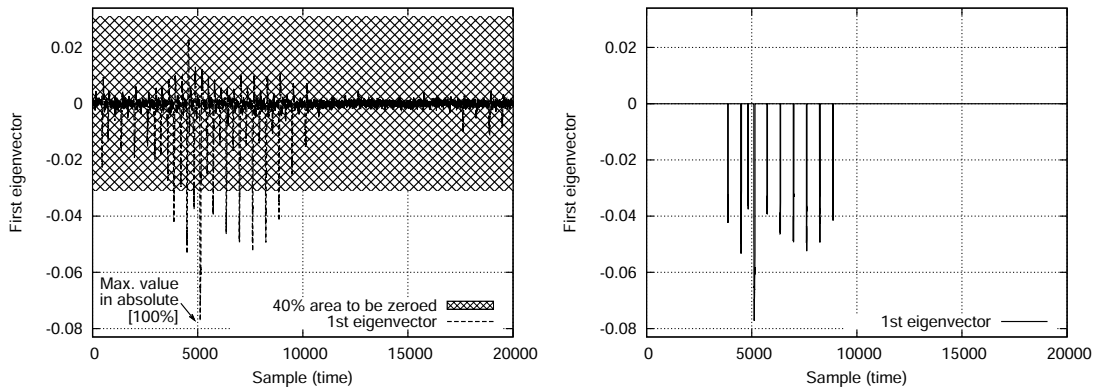
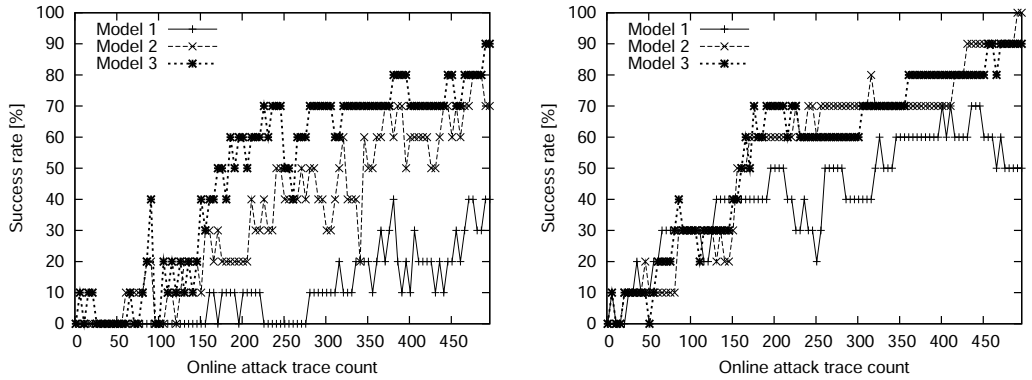


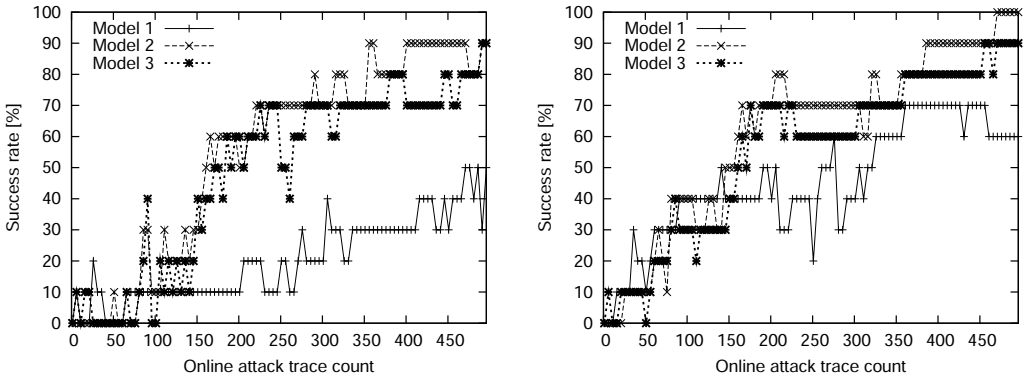
Figure H.1: Main eigenvector without thresholding (*left*), and the same with a 40% thresholding level (*right*).



(a) Without threshold.

(b) With threshold at 40 %.

Figure H.2: Success rate comparison between mono-partitioning models M1, M2 and combined model M3 for two different thresholds and 1,000 traces per class.



(a) Without threshold.

(b) With threshold at 40 %.

Figure H.3: Success rate comparison between mono-partitioning models M1, M2 and combined model M3 for two different thresholds and 32,000 traces in total for the training (to be divided between respectively 16, 2 and 32 classes).

H.2.3.3 Second Choice: Training-Limited Evaluation.

If we follow the first option, we take 32,000 traces in general. Thus, for a constant number of traces per class, we have 32,000/16 = 2,000 traces by class for model **M1** and 32,000/2 = 16,000 traces by class for **M2**. The combined model **M3** corresponds therefore to an amount of 32,000/32 = 1,000 traces by class. In this second case, we use systematically 32,000 for the training of all models M1, M2 and M3. As a consequence, model M2, that has the fewer number of partitions, will have its template evaluated more accurately than M1 and M3.

The two plots in Fig. H.3 show that the models combination does not so much gain on the attack. Indeed, the success rate of model M3 is very close to the success rate of the model M1.

H.2.4 Conditional Entropy

As explained above in Sec. H.2.1, the conditional entropy gives an idea about the robustness of the circuit, irrespective of any attack. The value of the conditional entropy tends to a limit value in function to the number of traces used for profiling [118]. For our experiment, we took a large number of traces during the profiling phase to have an approximation of this limit value. This will help us compare the circuit robustness against attacks using models M1, M2 or M3. Is our circuit very vulnerable against an attacker who combines model? The figure H.4 attempts to answer this question.

The use of PCA provides new directions corresponding to different eigenvectors. The number of these directions depends on the cardinality of the sensitive variable. For example, in this study, we have 15 directions for the model M1, 1 direction for the model M2, and 31 directions for model M3. The first direction summarizes a large percentage of variance of data. Making a comparison of robustness using only this first direction may seem satisfactory, but this study shows that the more directions, the greatest the estimated leakage (*i.e.* the smallest the conditional entropy). Combined models are thus

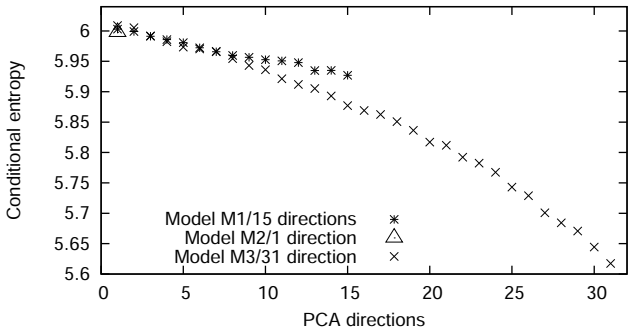


Figure H.4: Conditional entropy comparison between different models.

an opportunity to discover new leakage modes, as already noted for multi side-channel (power+EM) combination in [432]. This noting is actually a warning to the security evaluators: the robustness of an implementation can be underestimated if the models are either inappropriate (since incomplete, and thus should be completed with another or some other models) or contain too few partitions.

H.3 Combined Correlation Attacks

One difficulty for improving the side channel analysis or the template attack in presence of large noise is to identify the leaking samples, also called *Points Of Interest* (POIs). They correspond to the dates when the sensitive data is indeed processed and leaking the most. As already mentioned in the previous section when discussing the thresholding method, there is an obvious trade-off in the selection process for POIs. The more of them are selected, the more information is collected, but the more noise is kept. The difficult task consists in separating the signal from the noise.

Several techniques have been proposed to identify the POIs. The *Sum Of Squared pairwise (T-)Differences* (or sosd [141] and sost in [143]), the mutual information (MI [271]) and the *Principal Component Analysis* (PCA [13]) are four widespread examples. In this section, we study these methods and compare their efficiency, by applying them on two sets of measurements, one at short distance from the chip and another, one more noisy, at 25 cm from the chip. For these experiments we used a SASEBO-G board [391] embedding an AES hardware implementation. For these two sets of electromagnetic measurements $\mathbf{O}(t)$ we notice that a CPA can be successfully performed, by using the Hamming distance model between the penultimate and the last round state of the AES.

H.3.1 Techniques for Revealing the POIs

H.3.1.1 The sosd *versus* sost *versus* MI.

The computation of the sosd leakage indicator metric requires to average the traces in a given partitioning. In the original proposal [141], the partitioning concerns all the 256 values of an AES state byte. The SASEBO-G implementation is known to leak

the Hamming distance between the penultimate and the last round. Indeed, we succeed CPA for the both sets of measurements in this model. Therefore, we decide to restrict the values of the leakages to the interval $[0, 8]$, according to $\mathcal{L} = HW(\text{state}_9[sbox] \oplus \text{ciphertext}[sbox])$, where $sbox \in [0, 16[$ is the substitution box index. If we denote $o_i(t)$ all the samples (t) of the i^{th} realization of observation $\mathbf{O}(t)$, then the averages $\mu_j(t)$ in each class $j \in [0, 8]$ is given by the mean of set $\{o_i(t) \mid l_i = j\}$. Then their squared pairwise difference is summed up to yield the sosd.

The sost is based on the T-Test, which is a standard statistical tool to meet the challenge of distinguishing noisy signals. This method has the advantage to consider not only the difference between their means μ_j,$\mu_{j'}$ but as well their variability $(\sigma_j^2, \sigma_{j'}^2)$ in relation to the number of samples $(n_j, n_{j'})$. The definition of the sosd and sost is given below:

$$\text{sosd} \doteq \sum_{j,j'=0}^{8} \left(\mu_j - \mu_{j'}\right)^2 \qquad \text{and} \qquad \text{sost} \doteq \sum_{j,j'=0}^{8} \left(\frac{\mu_j - \mu_{j'}}{\sqrt{\frac{\sigma_j^2}{n_j} + \frac{\sigma_{j'}^2}{n_{j'}}}} \right)^2 .$$

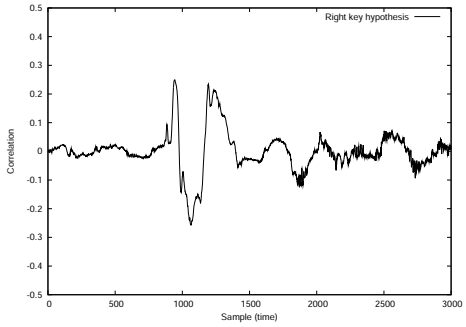
The sosd and the sost for the two EM observation campaigns are plotted in Fig. H.5. We notice that the correlation trace, the sosd and sost curves are matching for the measurement at 0 cm. But, although we use for the partitioning the same leakage function $\mathcal{L}$ and although we find the right key with a CPA on the measurement at 25 cm, the sosd curve does not highlight the right time sample, *i.e.* that where the key can be retrieved by CPA. This figure H.5 shows that the sosd metric is not always an efficient metric for revealing the points of interest. Indeed, we have tried to execute CPAs on the samples highlighted, but they all fail. Regarding the sost on the measurement at 25 cm, several POIs are revealed among samples that are not related to the secret data. Thus sost is neither a trustworthy tool to identify POIs.

Regarding the MI, also plotted in Fig. H.5, it matches well the sost at short distances, but features peaks with no information (notably the samples 441 and 975). It is thus not a reliable tool. The principal reason is that the PDFs are poorly estimated in the presence of large amounts of noise.

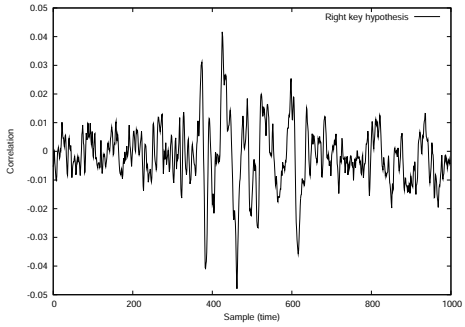
H.3.1.2 The PCA.

As previously explained in section H.2.2.2, the PCA aims at providing a new description of the measurements by projection on the most significant eigenvector(s) of the empirical covariance matrix of (μ_j). If we compare the success rate of the CPA, applied after a PCA, we can notice, that in the case of the campaign at distance, featuring a high level of noise, the eigenvector corresponding to the greatest eigenvalue is not necessarily suitable. The success rate of the CPA after a projection onto each of the nine eigenvectors is given in Fig. H.6. At 25 cm, we notice that the projection onto the first eigenvector is not necessarily the most suitable, since it does not yield the best attack success rate. The projection onto the third eigenvector turns out, quite surprisingly, to

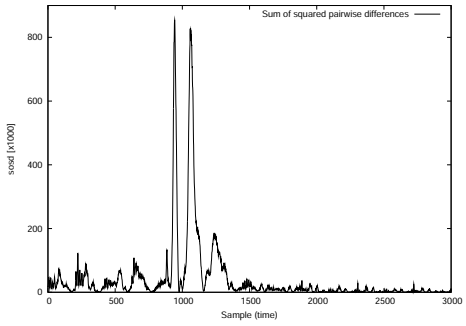
Correlation trace for campaign at 0 cm



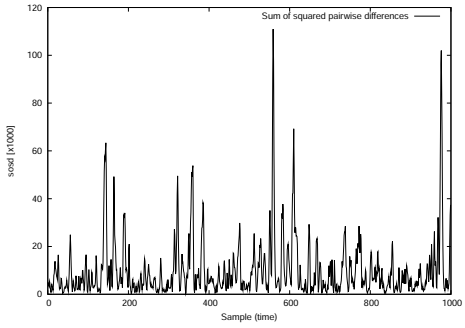
Correlation trace for campaign at 25 cm



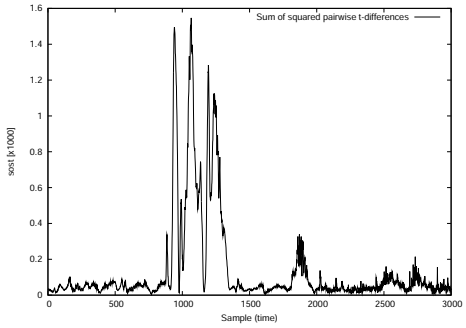
sosd for campaign at 0 cm



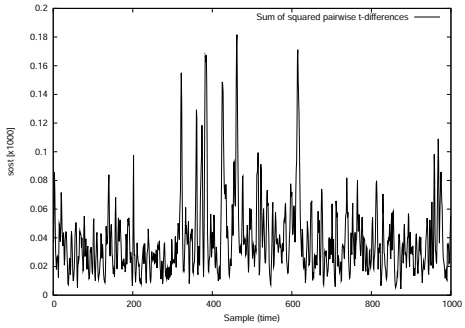
sosd for campaign at 25 cm



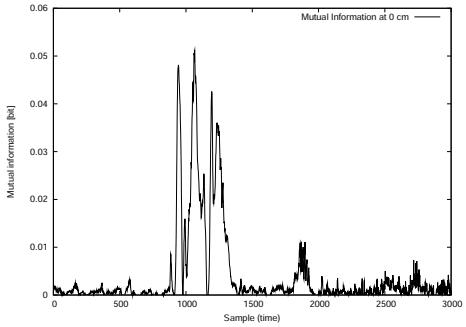
sost for campaign at 0 cm



sost for campaign at 25 cm



I(O;l) for campaign at 0 cm



I(O;l) for campaign at 25 cm

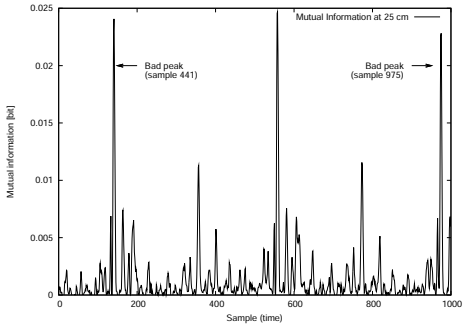


Figure H.5: Correlation traces, sosd, sost and MI obtained for the right key hypothesis.

be more efficient. At the opposite, when the noise level is low and the electromagnetic probe set at short distance, the projection onto the first vector is indeed more efficient.

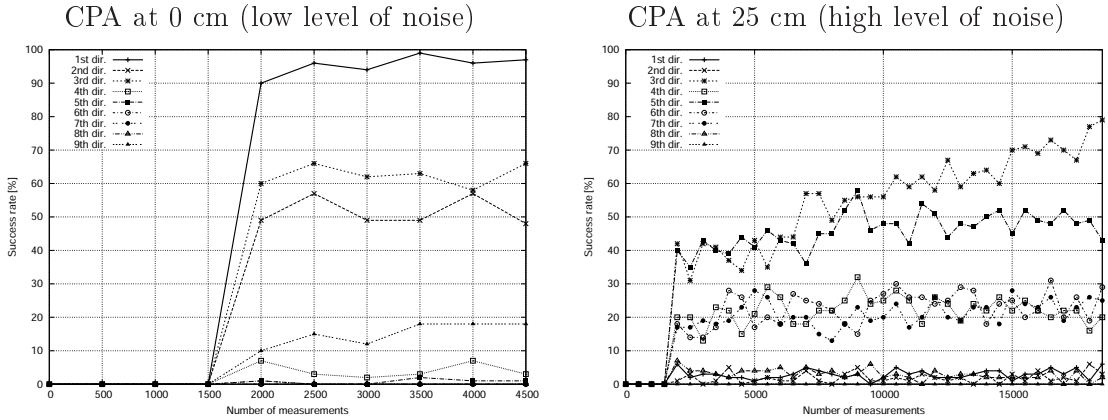


Figure H.6: Success rate of the CPA after PCA pre-processing.

This phenomena can be explained by the fact that the number of curves in the sub-set corresponding to the Hamming distances 0 and 8 are in same proportion, nevertheless the level of noise is higher, since they contain the fewest number of traces. Indeed, the proportion of traces available for the training is equal to $\frac{1}{2^8} \cdot \binom{8}{l}$, which is lowest for $l = 0$ or 8. The estimation of those classes is thus less accurate.

In order to improve the PCA, we have reduced the number of partitions from 9 to 7 sub-sets depending on the Hamming distance $HD \in [1, 7] = [0, 8] \backslash \{0, 8\}$. We observe that, under this restriction, the best success rate is obtained for the projection on the first eigenvector. In the meantime, the condition number of the empirical covariance matrix decreases, which confirms that the weakly populated classes $l \in \{0, 8\}$ added more noise than signal to the PCA. Amazingly enough, this approach is antinomic with the multi-bit DPA of Messerges [308]. If we transpose from DES to AES, Messerges suggests at the opposite to get rid of the classes $l = [1, 7]$ and to retain only $l = \{0, 8\}$. Those extremal samples have two ambivalent properties. They convey the most information, as shown in Tab. H.1, but also are the rarest samples, and thus are the most noisy coefficient in the covariance matrix. As Messerges does not make use of extra-diagonal coefficients, his attack is not concerned by this fact.

H.3.2 Combining Time Samples

H.3.2.1 Observations.

The correlation trace obtained for the right key with measurements at distance is given in Fig. H.7. We observe that the correlation traces are extremely noisy. Moreover for some time samples, identified in as Sample{1,2,3,4} in Fig. H.7, the magnitude of the correlation trace obtained for the right key is clearly higher than the magnitude of the correlation traces for bad key hypotheses. These samples are all located within the same clock period that corresponds to the last round of the AES. At the four identified dates,

Table H.1: Information and probability of the Hamming weight of an 8-bit uniformly distributed random variable.

Class index l	0	1	2	3	4	5	6	7	8
Information [bit]	8.00	5.00	3.19	2.19	1.87	2.19	3.19	5.00	8.00
Probability [%]	0.4	3.1	10.9	21.9	27.3	21.9	10.9	3.1	0.4

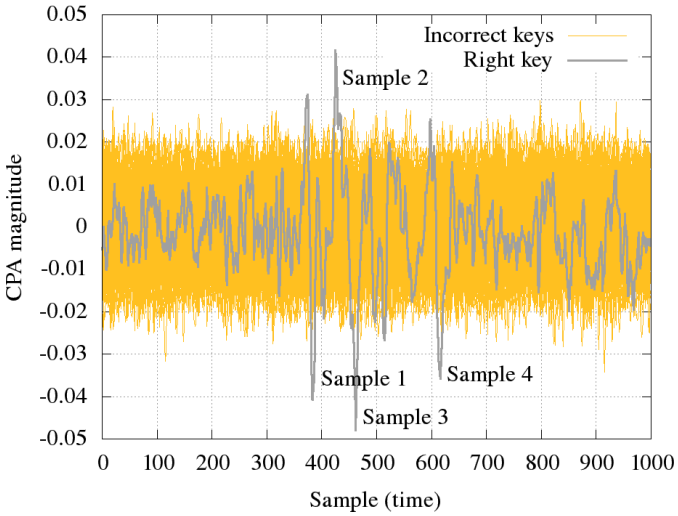


Figure H.7: Correlation traces obtained for the right key hypotheses and for incorrect key hypotheses at 25 cm.

the sample are undoubtedly carrying secret information.

H.3.2.2 Sample Combination Principle and Results.

We aim at showing that there is a gain in combining the leaks from the four identified dates. First of all, we confirm that the four samples of peak CPA are actually POIs. To do so, we perform successful CPAs at these time samples. The result is shown in Fig. H.8: all four attacks pass over a success rate of 50 % after 12,000 traces. Second, we devise a method to attack that exploits at once all those samples. Similar methods have already be introduced in the context of combining samples in order defeat masking countermeasures [365]. In [68], Chari *et al.* suggest to use the product of two leakage models. In [232], Joye *et al.* recommend to combine two samples with the absolute value of the difference. As in our case we intend to combine more than two samples, we resort to the product for the combination function. We apply it to Pearson empirical correlation coefficients $\hat{\rho}_t$, where t are the four identified dates. The new distinguisher we promote is thus:

$$\hat{\rho}_{\text{combined}} \doteq \prod_{t \in \text{Sample}\{1,2,3,4\}} \hat{\rho}_t \,. \tag{H.1}$$

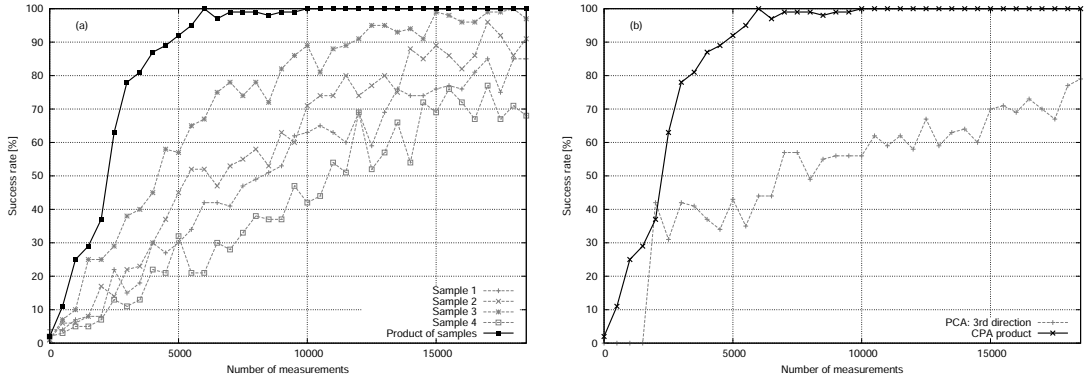


Figure H.8: (a)–*left*: Success rate of the mono-sample attack, and product of correlations attack; (b)–*right*: Comparison between a CPA using the pre-treatment by PCA and our product of correlation, introduced in Eqn. (H.1).

This technique applies well to the Pearson correlation coefficients, that are already centered by design. Thus it indeed puts forward the simultaneous coincidences of high correlation, while it demotes incorrect hypotheses for which at least one $\hat{\rho}_t$ is close to zero. As shown in Fig. H.8(a), the success rate of this new attack is greater than that for mono-samples attacks. Additionally, we confirm in Fig. H.8(b) that our combination defined in Eqn. (H.1), although simple in its setup, clearly outperforms a PCA after performing PCA.

However, we have only shown that when knowing some POIs in the curve, a powerful combining multi-sample attack can be devised. Now, for the time being, the only method to exhibit those POIs has been to apply a successful attack (a CPA in our case). Therefore, an open question is to locate those POIs without knowing the key beforehand or without conducting another less powerful attack. We suggest two solutions to spot the POIs: either online or by precharacterization on an open sample assuming the position of the POIs do not depend on the secret key.

H.4 Conclusion and Perspectives

In this paper, we have studied two examples of side-channel attacks combinations. The first contribution is the demonstration of a constructive multi-partitioning attack. We show that two partitioning can enhance the convergence of the success rate to one hundred percent; such attacks benefit from an exhaustive pre-characterization, since the number of templates increases, and that the training phase length is the product of the training phase for each partitioning. The second contribution is to highlight the existence of the leakage model in far field EM signals. We show how the leakage of each sample can be combined better than usual leakage reduction methods (*e.g.* the sosd, the sost or the PCA). This improvement comes from the fact each sample features a leakage of different nature that can be exploited individually, which is out of the reach of

global techniques that consist in identifying points with large variation. Our improved combining distinguisher consists in multiplying the Pearson correlation coefficients for several POIs. Although this attack leads to better success rates than other attacks using different state-of-the-art pre-processing, we do think it can still be enhanced by another method to identify the points of interest accurately even when the side-channel observations are extremely noisy. As a perspective, we intend to apply those ideas to an online only attack, typically the MIA.

Appendix I

Defeating Any Secret Cryptography with SCARE Attacks

Extended version of article [215]

Authors: *Sylvain Guilley, Laurent Sauvage, Julien Micolod, Denis Réal and Frédéric Valette*

Abstract

This article aims at showing that side-channel analyses constitute powerful tools for reverse-engineering applications. We present two new attacks that only require known plaintext or ciphertext. The first one targets a stream cipher and points out how an attacker can recover unknown linear parts of an algorithm which is in our case the parameters of a Linear Feedback Shift Register. The second technique allows to retrieve an unknown non-linear function such as a substitution box. It can be applied on every kind of symmetric algorithm (typically Feistel or Substitution Permutation Network) and also on stream ciphers.
Twelve years after the first publication about side-channel attacks, we show that the potential of these analyses has been initially seriously under-estimated. Every cryptography, either public or secret, is indeed at risk when implemented in a device accessible by an attacker. This illustrates how vulnerable cryptography is without a trusted tamper-proof hardware support.

I.1 Introduction

Most cryptanalyses require the knowledge of the attacked algorithm. It is thus tempting for a cryptographic engineer to protect an algorithm by keeping its specifications secret. In this case, the adversary faces the difficulty to attack a blackbox, hardly distinguishable from a random number generator. Only cube attacks succeed in these contexts, provided the cipher can be expressed as a polynomial of low degree.

As early as in the 1880s, Auguste Kerckhoffs [240] pleaded for cryptographic algorithms publication. The traditional approach to discourage algorithms secrecy is the lack of scrutiny, which could mean that original powerful attacks could be devised "out of the box" against the algorithm if it was disclosed. This risk has nowadays almost vanished, since many standardized and thoroughly studied algorithms exist. It is thus easy to mark down a reference algorithm to make it partially customized while maintaining its security level in the meantime. Therefore, the benefit is to dissuade any prospective opponent by adding an effort of algorithm-recovery prior to starting the key-recovery work. The reasoned usage of standard algorithm modification is thus safe from a computational point of view. However, this approach does not provide any improvement in terms of forward secrecy, since once the algorithm-recovery barrier is overcome, the security level is merely that of the underlying standard algorithm.

Another appeal of side-channel analyses (SCA) is their suitability to reverse-engineer an algorithm, a technique known as SCARE. However, although the side-channel attacks database [1] indexes more that 700 bibliographic references about SCA, we have found only 9 (namely [342, 344, 106, 128, 466, 9, 84, 374, 145], discussed later) that deal with SCARE. This low percentage of SCARE publications is certainly detrimental to the scientific progress on this topic. Most publications so far about SCA reverse-engineering (SCARE) concentrate on block ciphers, where in addition the plaintext can be chosen. In this article, we show that SCARE can be extended to any context where either the plaintext or the ciphertext is only known. We also demonstrate for the first time a SCARE attack on a stream-cipher and on a substitution permutation network (SPN) block cipher. It seems that the potential of SCARE goes much beyond what was previously expected.

Side-channel analyses are attacks that are virtually able to probe any node from a circuit after post-processing of a database of side-channel physical measurements. They make attack strategies such as that of Itai Dinur and Adi Shamir [113] (cryptanalysis using the sole knowledge of one bit amongst the first round state) possible, albeit in a statistical modus operandi. Therefore, we explicit how the techniques known so far, for example that exploit collisions, can be improved and gain generality.

The rest of the paper is organized as follows. Section I.2 recalls the public state-of-the-art about SCARE attacks. Sections I.3 and I.4 describe two new attacks on two representative blocks making up cryptographic algorithms. The first attack shows how the linear part of an unknown stream cipher can be easily recovered using only the knowledge of an initial value. This technique is practically illustrated on a stream cipher similar to those customarily used in RFIDs (called "RFID-like" in the sequel). The second one describes a known plaintext attack on an unknown non-linear function. This method which can be applied either to a Feistel or a SPN block cipher is demonstrated on a DES implementation which is publicly available. Finally, section I.5 concludes on the efficiency of our attacks and on further possible improvements. Further considerations about SCARE on stream and block ciphers are given in appendix I.6 and I.7 respectively.

1. Service hosted by the University of Boston: http://www.sidechannelattacks.com.

I.2 SCARE: State-of-the-Art

Physical attacks based on Side Channel Analysis (SCA) or on Fault Analysis (FA) target a secret usually manipulated by a algorithm with public specifications. SCA can also be used for Reverse-Engineering (SCARE) against implementations of a private algorithm.

I.2.1 State-of-the-Art about Reverse-Engineering of Secret Algorithms Embedded in a Device

Algorithms coded in software are exposed to an illegitimate access of their machine code. Indeed, as the memory is a separate component, it must be readable for its programme to be executed. Therefore, an attack strategy can consist in soldering the memories chips out, which can be done easily without damaging the component, in a view to drive it by a rogue processor that is going to dump the software instead of executing it. As a consequence, it is more secure to conceal the cryptographic algorithm into the cryptographic device. It has been believed for a long time that this was the definitive solution against its retrieval. The smartcards are typical examples of security products that enforce this idea.

I.2.2 Physical Attacks on Tamper-Proof Hardware

However, attackers became imaginative to read inside the secure chips. As of today, the reverse-engineering of an algorithm embedded in an electronic device can be done by various methods. The most straightforward one is simply to recover the layout by taking pictures of the different technological layers. This is not common practice but some specialized businesses can do so. It was recently illustrated on the example of the NXP MyFare 1k secure memory card, fabricated in an old CMOS technology [340]. In this example, the algorithm was hard-coded, hence its structure was easy to retrieve. However, simple counter-measures can be imagined. For instance, if the structure of interconnected logic gates can be retrieved by microscopy, the content of a non-volatile memory point (such as EEPROM or Flash) is not that easy to read-back optically [12]. Incidentally, it is a natural counter-measure if the algorithm is intended to be customizable.

FPGAs have become a viable alternative to ASICs due to their flexibility and their low cost for small to medium volumes. The functionality of most FPGA (with the remarkable exception of anti-fuse technologies) is stored in volatile memory points. Thus the read-back by invasive methods is also impossible, or at least very difficult. However, the SRAM points value is stored externally in a flash memory. But the high-end FPGA manufacturers (*e.g.* Actel, Altera, Xilinx) encrypt the memory's content. Consequently, bitstream encryption makes reverse-engineering as hard as decryption without the knowledge of the key.

No fault attack against embedded SRAMs (in ASIC or FPGA) aiming at a reverse-engineering is known: it does not seem trivial to deduce any information on a secret architecture by the result of its mutation, even if it is controlled. But another option to

retrieve a functionality is to fault the bitstream before it is loaded into the FPGA. Indeed, bitstreams are not well protected against DFA, even if they embed some redundancy, they remain malleable: they can be forged with a high success probability. The idea is not attack the customized parts, but to reduce the number of rounds, for instance, so as to ease the reverse-engineering.

I.2.3 SCARE Techniques

Physical attacks based on Side Channel Analysis (SCA) or on Fault Analysis (FA) usually target a secret manipulated by a public algorithm. SCA can also be used for Reverse-Engineering (SCARE) against implementations of a private algorithm. We can identify in the publicly available literature three techniques that have been proposed to reverse-engineer an algorithm.

The most natural idea is to use pattern matching techniques. Template Analysis for Reverse-Engineering is mainly used for instructions identification in embedded applications. The feasibility for Java card was shown in [466] and for microcontrollers in [128, 145]. The template matching can be optimized by coupling them with instruction sequence statistics.

A second approach is based on classical Correlation Analysis. It has been applied to public key cryptosystems [9] as well as on both hardware or software Feistel scheme implementations. For a software design, one-round intermediate values leak with a high enough signal-to-noise ratio for being guessed by SCA. Indeed, they are computed sequentially and stored in a register. The feasibility of a SCARE for a software DES implementation has been proved in [106]. For a hardware implementation, due to Feistel properties, an accurate side channel analysis permits to guess the output of the Feistel function for any chosen right half plaintext. Then, using interpolation methods, this unknown function can be recovered [374].

The last technique is based on collision analysis and was first applied on the private GSM A3/A8 algorithm where only the overall structure of the algorithm was publicly available. R. Novak proposed a strategy for identifying one confidential substitution table T_2 of this algorithm using collision [342]. Novak generalizes his attack using SDPA (Sign-based DPA) in [344], and summarizes his method in [343]. However, he needs as a prerequisite the knowledge of the secret key and of the confidential substitution sbox (sbox) T_1. This attack was improved in [84] (also refer to the preliminary version in [83]). Combining collisions and CPA principles, both key and sbox T_1 are found back. Then, Novak's attack is applied to discover the remaining confidential information about A3/A8.

I.3 SCARE on a Stream Cipher

Due to their small size, stream ciphers are often used in smartcard or low power IC. Contrary to block ciphers such as AES, stream ciphers are usually proprietary and dedicated to a specific application. The security of these systems is usually provided by

the secrecy of the algorithm. The reverse-engineering of the algorithm is often followed by a cryptanalysis of the system. It has been illustrated by famous examples such as the attacks on the GSM algorithm A5/1 [46] and more recently the cryptanalysis of the MyFare [135] which has followed the recovering of the algorithm CRYPTO1 by [339]. A similar story happened for the DECT Standard Cipher in early 2010 [341]. In this section, we will see how side channel techniques can be efficiently used to reverse-engineer an RFID-like stream cipher. In order to detail our technique, we will simplify our stream cipher by reducing it to a simple Linear Feedback Shift Register (LFSR) followed by some non-linear functions.

I.3.1 Stream Cipher Presentation

- j : The j^{th} experiment.
- t : The t^{th} clock cycle.
- i : The i^{th} bit of the shift register.
- $REG_t^j[i]$: The i^{th} bit of the register at the clock cycle t for the j^{th} experiment.
- IV_t^j : The bit of the IV being input in the register at the clock cycle t for the j^{th} experiment.
- $P[i]$: The i^{th} bit of the LFSR polynomial.
- K : The initialisation seed of the register. It is the same for all the experiments.
- L : The length of the register.
- F_t^j : The feedback at the clock cycle t for the j^{th} experiment.
- FIV_t^j : The feedback depending on IVs at the clock cycle t for the j^{th} experiment.
- FK_t : The feedback depending on the constant seed at the clock cycle t.
- $RADHYP[O]$: The radiation hypothesis on an object O.

Figure I.1: Notations.

The stream cipher implementation we study in this paper is an LFSR filtered by a non-linear function. The LFSR is an L-bit shift register initialized by a constant seed K that can be considered as the key. Usually, each ciphering j of the stream cipher needs a random initialisation vector noted IV^j used to make each ciphering independent from others ciphering produced by the same key. For each ciphering, at each clock cycle t a bit of IV_t^j XORed with a feedback F_t^j enters into the register. The feedback of the register can be represented as a polynomial P applied on the register. (When there is an XOR connected to a cell i of the register then $P[i] = 1$ else $P[i] = 0$.) The notations are summarized in the figure I.1.

The feedback F_t^j can be expressed by the following Eq. (I.1):

$$(F_t^j) = \left(\bigoplus_{k=0}^{k=L-1} P[k].REG_{t-1}^j[k] \right). \tag{I.1}$$

As the LFSR is linear, its equation can be written as the XOR of two equations, each one represented as an L bit register:

- The first register REG1 is initialized by the seed K. The IV value is null for each experiment. So we can notice that the LFSR has the same state behaviour for each experiment because the feedback only depends on the seed K.
- The second register REG2 is initialized by a seed equal to 0. The value which enters in the register is the IV_t^j so we can compute the state at each time t for each experiment j which depends only on IV.

We can express the feedbacks F_t^j as the XOR of the feedback FK_t and FIV_t^j. And from the Eq. (I.1), we express FK_t and FIV_t^j in Eq. (I.2) and (I.3).

$$(FK_t) = (\bigoplus_{k=0}^{k=L-1} P[k].REG1_{t-1}^j[k])\,. \tag{I.2}$$

$$(FIV_t^j) = (\bigoplus_{k=0}^{k=L-1} P[k].REG2_{t-1}^j[k])\,. \tag{I.3}$$

Particularly for FIV_t^j we can notice that for $t < L$ the feedback only depends on the t first coefficients of the polynomial. Indeed the initial state of the register is null and is shifted at each clock cycle. We can therefore simplify FIV_t^j by Eq. (I.4):

$$\forall t < L, (FIV_t^j) \quad = \quad (\bigoplus_{k=0}^{k=t-1} P[k].REG2_k^j[k])\,. \tag{I.4}$$

These equations will help us recover the LFSR characteristics L and P by Correlation ElectroMagnetic Analysis (CEMA).

I.3.2 Target Object for the Side Channel Analysis: Radiation Hypothesis

The first step of a CEMA is to find a relevant radiation hypothesis. In our case, we need to model the behaviour of a register. The current rise during the shifts in a register from '0' to '1' or '1' to '0' is much higher than the static dissipation observed for the holding state. If the low consumption is noted 0 and the high consumption is noted 1 then the consumption of the bit register can be approximate by the XOR between the old and the new value of the register. Table I.1 presents this radiation hypothesis for a shift register. A CEMA realized on this object will show when the two bits are manipulated. This classical model is usually known as the Hamming distance model introduced by [60].

For an LFSR, at each clock cycle t, $REG_t^j[i]$ is replaced by $REG_t^j[i-1]$. We can notice that the radiation hypothesis on $REG_t^j[i] \oplus REG_t^j[i-1]$ is equivalent to the radiation hypothesis on $REG_{t-1}^j[i-1] \oplus REG_{t-1}^j[i-2]$ and step by step to the radiation hypothesis $REG_{t-i+1}^j[1] \oplus REG_{t-i+1}^j[0]$.

In the feedback register, the first state bit corresponding to $REG_t^j[0]$ is replaced by the XOR between $IV^j{}_t$ and the feedback F_t^j.

Table I.1: Radiation hypothesis.

$\mathbf{REG}_t^j$**[i]**	$\mathbf{REG}_t^j$**[i-1]**	$\mathbf{REG}_t^j$**[i] xor** $\mathbf{REG}_t^j$**[i-1]**	**Modelized radiation**
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

So the radiation hypothesis on $REG_t^j[i] \oplus REG_t^j[i-1]$ is equivalent to $RADHYP[REG_t^j[0]]$ which can be written according to this expression:

$$RADHYP[REG_t^j[0]] = REG_t^j[0] \oplus IV_t^j \oplus F_t^j \,. \tag{I.5}$$

The problem on $REG_t^j[i]$ therefore boils down to a problem on the first state bit of the register. As done usually, we assume that the effect of a constant can be withdrawn. So we can simplify F_t^j and $RADHYP[REG_t[0]]$ by using Eq. (I.4).

$$RADHYP[REG_t^j[0]] = REG2_t^j[0] \oplus IV_t^j \oplus \Big(\bigoplus_{k=0}^{k=t-1} P_{t-1}[k].REG2_k^j[k]\Big) \,. \tag{I.6}$$

The radiation hypothesis on $REG_t[0]$ only depends on t first bits of the secret P.

Figure I.2: Theorem 1.

From Eq. (I.6) we can announce the Theorem 1, displayed in Fig. I.2.

I.3.3 Recovering LFSR Characteristics

From the radiation hypothesis and j experiments, it is possible to recover the LFSR characteristics (L and P) by CEMA. This recovery is divided in two parts:

1. the search of the register length,
2. the search of the polynomial value.

According to Theorem 1 (figure I.2), for $t = 0$ and $i = 0$, the radiation hypothesis $RADHYP[REG_0^j[0]]$ only depends on IV_0. So computing a CEMA on this value let us find the length of the register. Indeed IV_0 is shifted in the register before coming out of it. The number of clock cycles with CEMA peaks indicates the times where IV_0 is shifted and consequently the length L of the shift register.

To find the polynomial P an application of the Theorem 1 (figure I.2) by recurrence is done. For each step $0 < t < L$, $P[0, t-1]$ is assumed known. We want to guess $P[t]$. Two CEMAs with $P[t] = 0$ or $P[t] = 1$ permit to check if an XOR is present on $REG[t]$. Step by step it is then possible to find the polynomial by induction. The procedure is

illustrated in figure I.7 (page 263). For an L bit register, we need to perform L CEMAs to recover the polynomial P. So this attack is linear in the size L of the register.

I.3.4 Practical Attack

To validate this attack we have implemented a stream cipher in an Altera Stratix FPGA. FPGAs are usually available on the market in thiner technologies than ASICs because the former have the greatest need for improved integration capabilities. Therefore, their elementary logic is consuming and radiating less than ASICs; however, given that they are reconfigurable, any function implemented with user-logic (look-up-tables and commutation/switch matrices) in FPGAs require approximately thirty times more logic than in an hardwired ASIC [254]. Therefore the FPGAs also have an exacerbated side-channel leakage. All in one, the cryptographic designs mapped in FPGAs are often considered more vulnerable to observation attacks than those optimized for ASICs; they however represent a "worst-case" with respect to ASIC, which makes them a good target for security prototyping. The studied stream cipher, represented in Fig. I.6, is built with a 32-bit register ($L = 32$) and with 4 non-linear functions called F1, F2, F3 and G. When the stream is on progress, a flag is set in order to synchronise electromagnetic measures. We record 10000 electromagnetic measures ($0 < j \leq 10000$) of 40 clock cycles t with known randomized IVs. The first and the last significant peaks correspond to the flag: stream on progress. The 40 other peaks in the middle correspond to the 40 shifts of the register. The figure I.3(a) gives an example of these electromagnetic measures.

From the 10000 electromagnetic measures, CEMAs give the following results figure I.3(b). For the true assumption, CEMA peaks appear and for the wrong one we observe noise. On this figure we can observe that the CEMA leaks during 32 clock cycles corresponding to the length L of the register. From these electromagnetic measures we succeed in recovering the LFSR characteristics.

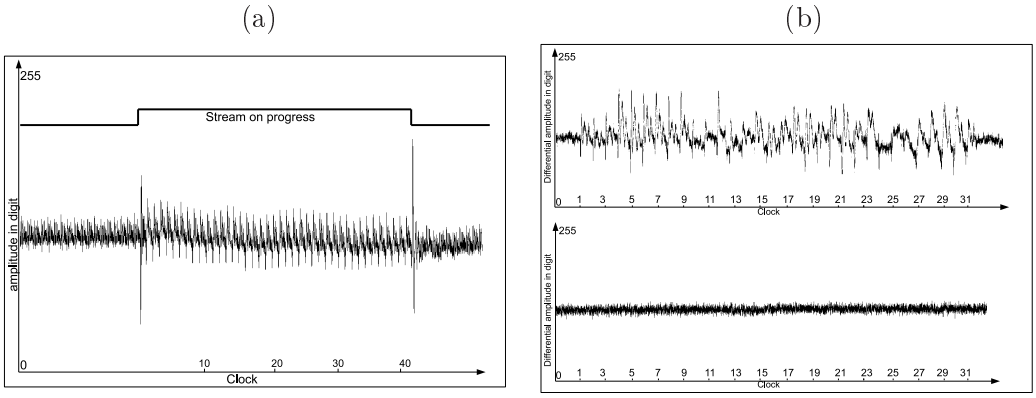


Figure I.3: (a) Electromagnetic measure of the stream cipher and (b) CEMA on the register.

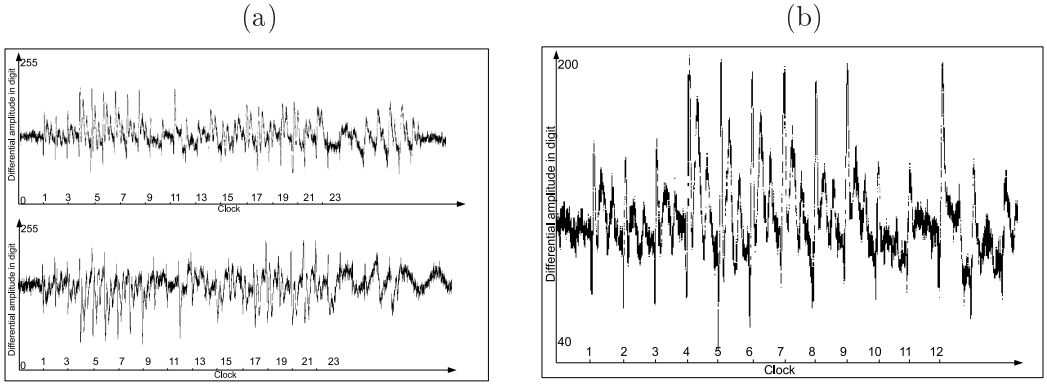


Figure I.4: (a) Key influence on CEMA results and (b) zoom in CEMA traces.

I.3.5 Further Analysis of the SCA Results

We showed earlier how the main characteristics of the LFSR can be easily recovered by SCARE techniques. Looking more precisely at the obtained curves, we can deduce more information leading to a full reverse-engineering of the algorithm. First of all, we can observe as shown in Figure I.4(a) that the peaks can be on the top or on the bottom of the curve secondly we can also observe that the peaks are quite different from one clock to another as shown in figure I.4(b).

The first difference is a horizontal symmetry due to the feedback constant value. In the previous CEMA, we assume that this constant value has no influence, which was actually not exact. In fact, this value can exchange the two considered sets of a CEMA. The obtained curve will be different if this value $FK_t \oplus FK_{t+1}$ is equal to 1 or equal to 0. By observing the curves we can easily deduce whether or not the constant value $FK_t \oplus FK_{t+1}$ is equal to $FK_r \oplus FK_{r+1}$. In the example shown in Figure I.4(a), we see that the curves are not identical but symmetrical and we can deduce that $FK_0 \oplus FK_1 = 1 \oplus (FK_1 \oplus FK_2)$ hence $FK_0 = 1 \oplus FK_2$. Doing this observation at each step i from 0 to n will yield n linear equations on FK_t what directly gives linear equations on K_t and allows to retrieve two complementary possibilities for the seed K.

The second difference remains in the size of the peaks at each clock cycle. As we can see on Figure I.4(b), the first three peaks are smaller than the next six peaks. Then we observe two small peaks again followed by a larger one and finally two small peaks. To sum up, we obtain small peaks at cycles $\{1, 2, 3, 10, 11, 13, 14\}$ and large peaks at cycles $\{4, 5, 6, 7, 8, 9, 12\}$. The values identified for the large peaks are very closed to the bits that are used in the entry of the non-linear functions. This can be explained by the fact that when the bit changes in the register, it also activates the wires and the logic gates used to compute the non-linear functions. We can note that all the entries of the non-linear functions produce a large peak but we can observe that "ghost peaks" [60] also appeared for example at clock cycle 6. This fact is classically caused by non-modelled biases between the side channel indicator and the implemented objects. Nevertheless, it

allows us to reduce the number of possible entries of the non-linear functions.

We now have enough information to perform the full reverse-engineering. Indeed, we know all the information on the LFSR (its initial value and its feedback polynomial) and we also have potential entries for the non-linear functions. To retrieve this function, we can apply the technique presented in the next section.

I.4 SCARE on Non-Linear Functions

All previous publications about SCARE rely on the fact that the message can be chosen. In some of these publications, the key must be known. In others, it shall not be known, however it must remain constant. We will assume that neither the plaintext nor the key is chosen. Additionally, in the proposed methodology, the key can change at every encryption without making the analysis fail.

The specificity of SPN operating at one round per clock period is that the future state cannot be partially known, unlike in Feistel schemes. We illustrate in this section a method to recover the non-linear functions (called sboxes in short) that applies both to SPN and Feistel ciphers. Also, given that confidential algorithms are usually not reconfigurable, it is unlikely to build a precharacterized database of typical SCA signatures. Thus, as no training phase is possible, template [69], stochastic [407] and MIA [141] attacks cannot apply. Under some conditions, called EIS (Equal Images under different Sub-keys), the profiling can be done without changing the secret element. Although this property can be used to characterize devices where the secret element is a key that is injected in a linear way into the algorithm [407], it is of no help to exploit non-linear functions. Therefore, our analysis must rely solely on correlation attacks with an assumed leakage model.

I.4.1 SCARE Attack Path

This section briefly indicates that an attack path can be defined in SCARE just as in SCA. We assume that the implementation under analysis is in synchronous logic, which means that the hardware resources split into two categories: registers, that evolve all concomitantly at the clock frequency, and combinatorial gates, that evaluate independently at a data-dependent pace. In this context, the easiest resource to identify are the registers. In the mainstream CMOS technology, their leakage is indeed very well captured by the Hamming distance model, already evoked in the Section I.3. It means that the number of bits that change in two consecutive values is present in the side-channel emanations. This behavior has been initially underlined for ASICs in [60] and for FPGAs in [435].

The rationale behind our SCARE attack relies on the *identification of transitions* that are predictable and full [435]. The methodology is sketched as follows. We make an architecture hypothesis; if it leads to a significant side-channel signature, it means that the hypothesis is correct. Let us assume for instance that the block encryptor to reverse-engineer is a Feistel scheme. The transition $L_0 \rightarrow L_1$ should yield a clear signature in

DPA. Now, $L_0 \oplus L_1 = L_0 \oplus R_0$ is independent of the unknown Feistel function. If there is actually one peak, we can validate that:

- The cipher is a Feistel network;
- The Feistel structure is simple (no multiple and/or cross Feistel with quarters of datapath, etc.).

Additionally, we get the following precious information to guide the rest of the analysis:

- What is the side-channel leakage amplitude of 32-bit transitions?
- Where in time is the first round?

The correlation will be $\sqrt{32} \approx 5.66$ times smaller for a mono-bit SCARE attack [437], and the attack can focus on the first clock period: in this example, the peak is maximum at sample 5745. Also, the attacker can take advantage of this signature to tune the empirical parameter ϵ introduced in [258] to make up for the division by abnormally small values.

At the opposite, if no signature is obtained for the $L_0 \rightarrow L_1$ transition, we would conclude that the cipher consumes the whole datapath in one single round. This is typically the case of SPN-type structures.

We do not address in this paper the complete attack path for arbitrary algorithms structures, because exotic features can lead in practice to *ad hoc* attack strategies. Instead, we focus on one blocking point: the reversal of sboxes when only their input or their output is known. In previous attacks, this was done on Feistel schemes, where both inputs and outputs can be controlled individually because the datapath splits into two halves. We do not make this assumption in our attack.

We illustrate the attack on a DES-like cipher, where the sboxes would have been customized. In particular, we use the side-channel measurements of the DPA contest [445], in a view to present results reproducible by our peers. As a matter of fact, the sboxes to retrieve are actually the genuine ones from DES, we pretend not to know. However, we based our attack on the sole knowledge of the right half of the datapath, so as to capture an attacks that could be transposed to SPN ciphers as well.

I.4.2 Brute-Forcing Sboxes

The attack presented in this section is a brute force retrieval of the sboxes. One vectorial Boolean function $n \rightarrow m$ can be expressed:

- component-wise (*i.e.* as m different Boolean functions sharing the same n inputs),
- using a reduced fanin, thanks to the recursive application of the identity for a Boolean function f: $f(a,b) = \overline{a} \cdot f(0,b) + a \cdot f(1,b)$. At each recursion, the fanin (*i.e.* the number of free variables) is reduced by one.

The attacker has the flexibility to choose the fanin i and the fanout o. In the sequel, we illustrate the case of $i = 2$ and $o = 1$. The complete list for all the $\{0,1\}^2 \mapsto \{0,1\}$ functions to retrieve is given in Tab. I.2. The terminology is straightforward for most gates; when a gate name ends with `A` or `B`, it means that the corresponding input is inversed prior to entering the function. For instance, $\texttt{AND2B}(\mathrm{A},\mathrm{B}) = \texttt{AND2}(\mathrm{A},\overline{\mathrm{B}}) = \mathrm{A} \cdot \overline{\mathrm{B}}$.

With respect to a canonical attack flow, not all the side-channel traces are processed simultaneously. More precisely, the first step consists in generating a whitelist of traces

that keep $6-i$ bits of the target sbox to a constant value (which is possible since the plaintext is known). Thereafter, a correlation program is called, with 16 weighting functions corresponding to the $2 \rightarrow 1$ sub-tables to guess (Tab. I.2) and with the traces restricted to the previously generated whitelist. As usual, when few traces are available, it is better to resort to CPA rather than to DPA [60].

The overall process consists in testing all the possible sub-functions of the sboxes, with more or less important restrictions on the inputs (here we keep $i = 2$ free bits) and one or all outputs components (here we illustrate a bit-by-bit component retrieval: $o = 1$). Therefore, to recover one sbox of DES (6 bits $\rightarrow$ 4 bits), $\binom{6}{6-i} \times 2^{6-i} \times \lfloor \frac{4}{o} \rfloor = 960$ CPAs are necessary. The choice for i and o makes it possible for the attacker to explore some trade-offs: the larger i, the more hypotheses to distinguish, but the more traces are suitable for the analysis (because the whitelist contains a ratio of $\frac{1}{2^{6-i}}$ of the available traces); However, the size of the whitelist does not depend on the value of o. Instead, the larger o, the more hypotheses in competition, hence the smaller the noise margin to tell which one is the best.

For the sake of illustration, we take an example depicted in Fig. I.5. In this figure, the correct function associated with:

- `sbox_num` = `0x0u`, (the sbox index, in [`0x0u`,`0x7u`])
- `sbox_mask_in` = `0x39u`, (the sbox inputs bits that are constant)
- `sbox_mask_in_val` = `0x11u`, (the value of the constant input bits)
- `sbox_mask_out` = `0x2u`, (the output bits guessed simultaneously)

is `0b1011u` = `0xbu`, hence the `OR2B` function. By definition, $i \doteq 6 - |\texttt{sbox\_mask\_in}|$ and $o \doteq |\texttt{sbox\_mask\_out}|$. In the figure, x_0 and x_1 are free variables and y_0 is the bit whose activity is predicted in order to retrieve the i-bit $\rightarrow$ o-bit function $DES_0(0, 1, 0, x_1, x_0, 1)[2]$.

One whitelist (screening of traces that keep $6-i$ bits constant for the targeted sbox) can serve for all the 4 components of the sbox. For instance, the four functions to be found in Fig. I.5 are:

1. for `sbox_mask_out` = `0x1u`: `0b1000u` = `0x8u`, hence `AND2`,
2. for `sbox_mask_out` = `0x2u`: `0b1011u` = `0xbu`, hence `OR2B`,
3. for `sbox_mask_out` = `0x4u`: `0b0110u` = `0x6u`, hence `XOR2`,
4. for `sbox_mask_out` = `0x8u`: `0b1101u` = `0xdu`, hence `OR2A`.

To be completely accurate, we shall recall that DES has an expansion permutation E before the key mixing. Therefore, the whitelist constrains the two neighbor sboxes in addition to the targeted one. However, the CPA selection function concerns only the o outputs of the sbox. Forcing some bits of the neighbor sboxes or having some of the $6-i$ active ones shared with them will therefore only affect the structure of the algorithmic noise generated by the $32-o$ bits of the R register. The impact of E is thus quasi-transparent for the SCARE attacks.

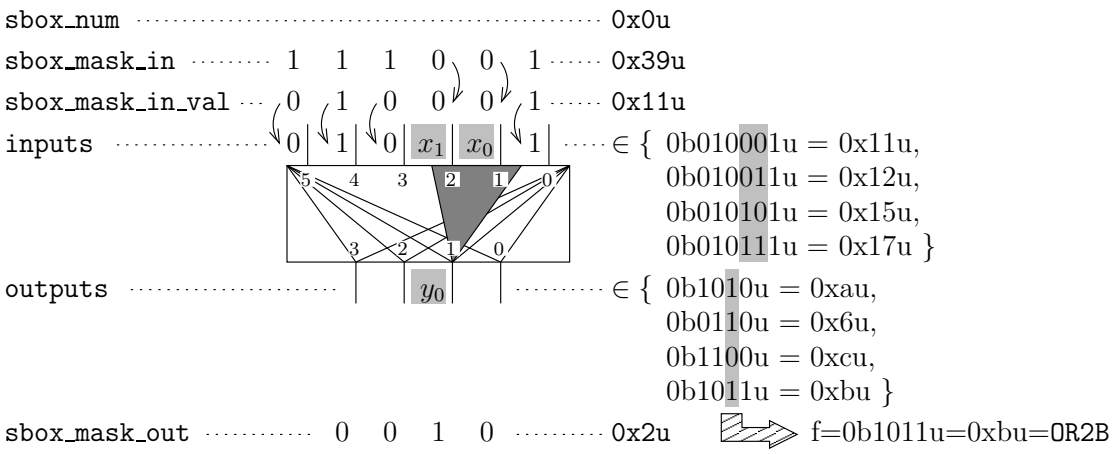


Figure I.5: Parameters `sbox_n`, `sbox_mask_in`, `sbox_mask_in_val` and `sbox_mask_out` that allow for the test of all possible sub-functions of the sboxes array of DES.

I.5 Conclusion and Perspectives

Side-channel attacks have been widely studied as a tool to perform key extractions. However, it can be used for other analyses, such as circuits on-line testing or architecture reverse-engineering. We illustrate in this paper that side-channel attacks can be used effectively to reverse-engineer secret algorithms. A practical reverse-engineering of a filtered LFSR is illustrated. In addition, a known-plaintext or known-ciphertext only attack is shown on a SPN or on a Feistel scheme. Those two new attacks demonstrate that SCARE is a technique able to effectively defeat secret cryptography. From the complexity point of view, those SCARE attacks, like the classical side-channel attacks, work whatever the size of the secret to recover. The reason is that they decompose the problem at the bit-level, where an exhaustive search amongst the hypotheses space is possible.

The concrete experiments confirm the efficiency and the practicability of our attacks as we were able to retrieve the full secrets we were looking for. We admit that unexplained correlations were observed, but they have a small impact on the success of our attack. Moreover, it helps us to recover more secret values than we initially though. Indeed, the initial state of the LFSR can be recovered too using our technique. Nevertheless, to precisely understand these observations, we need to model further the radiation of the full algorithm and specific implementation details. We envision this formal model to define precisely the required number of measurements needed for the attack to be successful. Also, we endeavour to apply those attacks on customized version of modern stream ciphers (*e.g.* Grain, Trivium or Mickey).

I.6 Appendix 1: Further Considerations about SCARE on a Stream Cipher

The detail of the algorithm used to reverse-engineer the stream cipher presented in Sec. I.3.3 and depicted in Fig. I.6 is given below.

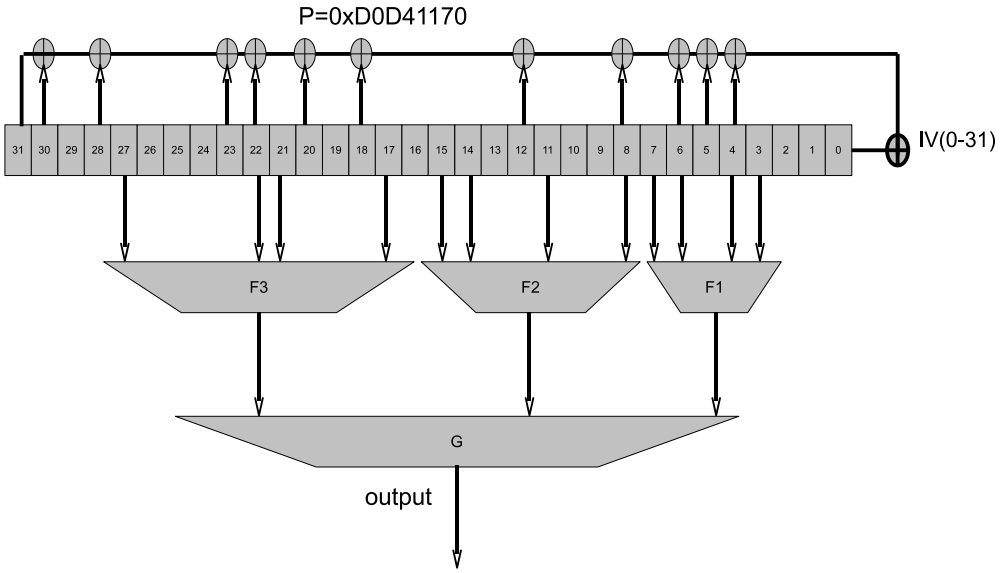


Figure I.6: Stream cipher implementation.

I.7 Appendix 2: Further Considerations about Brute-Force SCARE on Sboxes

I.7.1 Comparison of DPA versus SCARE

In this section, we carry out a theoretical comparison between DPA and SCARE for n to 1 Boolean functions. We recall the naming of those functions for $n = 2$ is provided in Tab. I.2. This discussion can extend without changes to multi-fanout sub-functions.

The DPA consists in computing the covariance between the curves and a power model. In the ideal case, the curves perfectly contain the power model of the few gates under attack plus a random noise. The attacker thus obtains a collection of: $\mathrm{Cov}(\mathrm{model}_0(k_0), \mathrm{model}_0(k))$; The attacker's goal is to find the correct k_0 amongst the k that are possible. In the usual setup where the key is incorporated into the datapath with a group operation $\oplus$, and where the confusion in implemented by an sbox S, the model_0 is: $x \mapsto S_0(x \oplus k_0)$. The DPA problem consists in finding: $\arg\max_k \sum_{x=0}^{2^n-1} (-1)^{S_0(x\oplus k_0)\oplus S_0(x\oplus k)}$. Of course, the solution is $k = k_0$ (see demonstration in [196]). The difficulty of the problem can be summarized by the noise immunity required to distinguish the auto-correlation

- Input
  - IV_t^j with $0 < j \leq 10000$ and the corresponding EM traces
  - L the register length
- Output
  - The feedback polynomial P
- Step 0: is there an XOR on REG[0]?
  - $P \leftarrow 0x00000000$
  - compute CEMA$_1$ on $REG_1^j[0] \oplus REG_1^j[1] \oplus P[0].REG_0^j[0] = IV_1^j$
  - compute CEMA$_2$ on $REG_1^j[0] \oplus REG_1^j[1] = IV_0^j \oplus IV_1^j$
  - if CEMA$_1$ leaks and CEMA$_2$ doesn't
  - $P \leftarrow P \mid 0x00000001$
  - else if CEMA$_2$ leaks and CEMA$_1$ doesn't
  - $P \leftarrow P$
  - else
  - Return Error
  - endif
- Step $t < L$: is there an XOR on REG[t]?
  - $P[0, t-1]$ is already known
  - compute CEMA$_1$ on $REG_t^j[0] \oplus IV_t^j \bigoplus_{k=0}^{k=t} P[k].REG_{t-1}^j[k] \oplus IV_0$
  - compute CEMA$_2$ on $REG_t^j[0] \oplus IV_t^j \bigoplus_{k=0}^{k=t} P[k].REG_{t-1}^j[k]$
  - if CEMA$_1$ leaks and CEMA$_2$ doesn't
  - $P \leftarrow P \mid (1 << t)$
  - else if CEMA$_2$ leaks and CEMA$_1$ doesn't
  - $P \leftarrow P$
  - else
  - Return Error
  - endif
- Return P

Figure I.7: SCARE algorithm to retrieve P.

Table I.2: Name of the $2 \to 1$ functions.

Index of f	$f(\mathbf{B}, \mathbf{A})$				**Boolean equation**	**Name**
	$f(1,1)$	$f(1,0)$	$f(0,1)$	$f(0,0)$		
0	0	0	0	0	0	Zero
1	0	0	0	1	$\overline{\mathrm{B} + \mathrm{A}}$	NOR2
2	0	0	1	0	$\overline{\mathrm{B}} \cdot \mathrm{A}$	AND2B
3	0	0	1	1	$\overline{\mathrm{B}}$	NOTB
4	0	1	0	0	$\mathrm{B} \cdot \overline{\mathrm{A}}$	AND2A
5	0	1	0	1	$\overline{\mathrm{A}}$	NOTA
6	0	1	1	0	$\mathrm{B} \oplus \mathrm{A}$	XOR2
7	0	1	1	1	$\overline{\mathrm{B} \cdot \mathrm{A}}$	NAND2
8	1	0	0	0	$\mathrm{B} \cdot \mathrm{A}$	AND2
9	1	0	0	1	$\mathrm{B} \overline{\oplus} \mathrm{A}$	XNOR2
10	1	0	1	0	A	A
11	1	0	1	1	$\overline{\mathrm{B}} + \mathrm{A}$	OR2B
12	1	1	0	0	B	B
13	1	1	0	1	$\mathrm{B} + \overline{\mathrm{A}}$	OR2A
14	1	1	1	0	$\mathrm{B} + \mathrm{A}$	OR2
15	1	1	1	1	1	One

from the non-trivial cross-correlations: $\sum_x (-1)^{S_0(x) \oplus S_0(x \oplus \epsilon)}$, $\epsilon \neq 0$.

The SCARE method inspired from the DPA will also use a covariance as an indicator (sometimes called "oracle") to distinguish the correct sboxes guess from the incorrect ones. In this respect, the SCARE attacker will compute similar quantities as for the DPA: $\mathrm{Cov}(\mathrm{model}_0(k_0), \mathrm{model}(k_0))$; In the same setup as described for DPA (group addition of the key and usage of an sbox S_0), SCARE consists in solving: $\arg\max_S \sum_x (-1)^{S_0(x \oplus k_0) \oplus S(x \oplus k_0)}$. Now, an equivalent problem is: $\arg\max_S \sum_x (-1)^{S_0(x) \oplus S(x)}$. Under this form, it appears clearly that **DPA is a particular case of SCARE**, where the set of possible Sboxes to retrieve can be written $S = S_0 \circ \tau_k$ (with τ_k the translation of vector k).

The similarity between DPA and SCARE arises from the fact that:

- the **key mixing** operation just precedes
- the **substitution boxes** layer.

Therefore, the mode of operation of DPA and SCARE target the same transition: $x \to S(x \oplus k)$, where:

- k is the unknown in the case of DPA, whereas
- S is the unknown in the case of SCARE.

Consequently, the attack strategy for SCARE is identical to that of the DPA. For instance, when analyzing a Feistel scheme such as DES, the sensitive transition targeted by both DPA and SCARE will be $R_0 \to R_1$ when attacking the first round, but $L_{15} \to L_{16}$ when attacking the last round.

The space to explore for finding a maximum is of size:

– $\#k = 2^n$, in the case of the DPA, to be contrasted by the
– $\#S = 2^{2^n}$ n-input Boolean functions.

It is thus more likely that DPA key extraction exhibits a larger signal-to-noise ratio than SCARE sbox extraction.

I.7.2 Specificity of SCARE w.r.t. DPA

Some issues we encountered during the attack are also specific to SCARE. We discuss in this section two of them.

Although a DPA tool can safely decide which sub-key hypothesis is the correct one by selecting the curve with the maximal signal in absolute value, it happens that this selection entails incorrect results in SCARE. Indeed, the correlation with f and with $\overline{f}$ yields exactly opposite results if f is balanced. This noting does not bother either DPA or CPA analyses on sboxes: as the sum of all differential peaks is equal to zero, which means that the second peak after the correct one has a lower amplitude (in absolute value), given that there is only one peak of maximal amplitude. Said differently, SCARE correlation curves for all the hypotheses have a lower contrast than their DPA counterparts. Concretely speaking, this has not been problematic in SCARE for the power measurements we used: power signal is always positive in case of one bit-flip. However, in EMA, this is not true: a bit-flip can be transducted as a negative measurement bias. Now, we assume that an electromagnetic sensor, even if meant to realize near-field side-channel acquisitions, does maintain a constant polarity over the spied sbox. In this case, the complete sbox can be retrieved (up to an unknown bitwise negation), with both a power and an EM analysis. However, the fact the both f and $\overline{f}$ compete as candidate Boolean functions has led to an attack failure, when targeting `sbox_num`: `0x0`, `mask_in`: `0x0f`, `mask_in_val`: `0x00`, `mask_out`: `0x4`. In happens that the highest peak in relative value appears in a secondary bounce, as shown in Fig. I.8.

Another peculiarity we came upon is that the automatic sbox recovery on the DPA contest acquisition campaign fails in the two cases:

1. `sbox_num` = `0x0`, `mask_in` = `0x1e`, `mask_in_val` = `0x02`, `mask_out` = `0x2`,
2. `sbox_num` = `0x0`, `mask_in` = `0x3a`, `mask_in_val` = `0x02`, `mask_out` = `0x2`.

Here, we observe `0x0` although the correct function is `0x2`. For these attacks, we get only four values of correlation: ± 15 % for 2×2 guesses, and ± 5 % for the 2×6 others. In both cases, we notice that the correlation model $\{-1, -0.5, 0, +0.5, +1\}$ is not satisfied. The singular correlation values are shown in Fig. I.9. To contrast this attack with a non-pathological one, we also give the correlations obtained for a complete $2 \rightarrow 4$ function recovery in Fig. I.10.

Finally, in Fig. I.9, we observe that the bounces do not have always the same period (for all the hypotheses), especially for `mask_out` = `0x1` and `0x4`. This seems as unphysical as the amplified bounce of `mask_out` = `0x4`. Those artifacts represent SCA second-order effects that make the analyses somehow subtle.

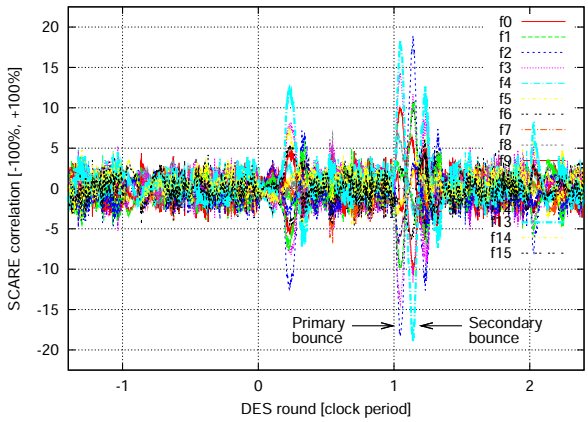


Figure I.8: The correct function is 13, but the complementary 2 is found instead with both guesses in absolute or in relative value.

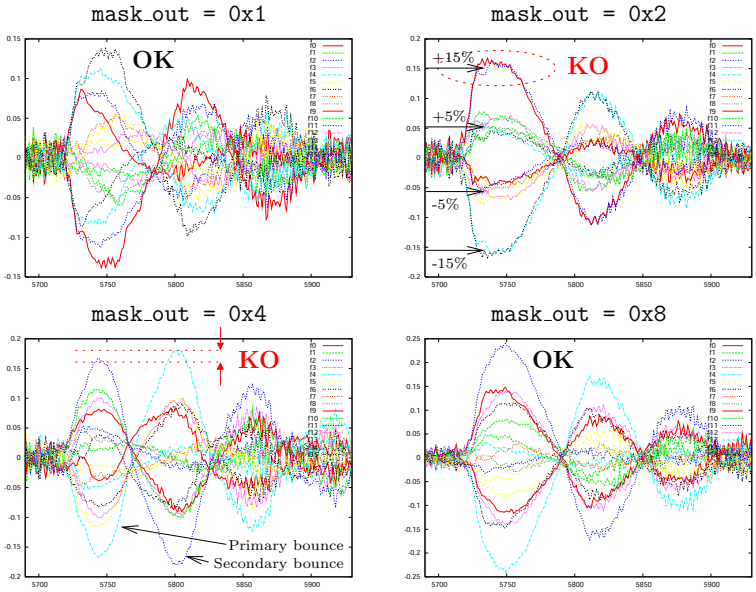


Figure I.9: Correlation traces obtained when retrieving the four Boolean components of the sbox #1, with input mask `0x1e`, and value `0x02`.

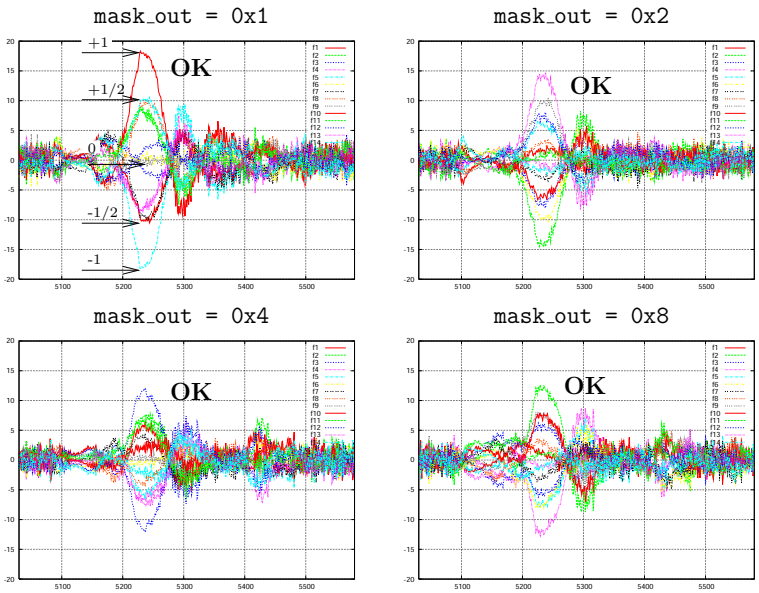


Figure I.10: Correlation traces obtained when retrieving the four Boolean components of the sbox #1, with input mask `0x33`, and value `0x00`.

I.7.3 SCARE on DES Sboxes Results

The figure I.11(a) presents the correlation coefficients obtained for the brute-force reverse-engineering of the sbox #7 of DES. We have indicated in green the second largest peaks. It can be seen that sometimes, the first and second largest correlations have about the same value. This is due to the (yet unexplained) degenerescence already observed in Fig. I.9 for `mask_out = 0x2`.

The shape of CPA curve in Fig. I.11(a) is in two parts. The explanation for this phenomenon is that, for this acquisition campaign, two bits correlate better than the two others. This is indeed put forward in Fig. I.11(b). The reason for the sbox output bits to correlate differently is a priori unknown, since they are treated in a similar way by the permutation layer of DES. The difference might be due to the acquisition conditions.

The two problems mentioned in the previous section are not fatal. By restricting the temporal window for CPA, the secondary negative bounce greater than the actual first one can be trivially filtered out. Indeed, as $(960-2)/960 > 99\%$ of the CPAs succeed in $t_0 \approx 5750$, it is easy to infer that there is something fishy about the 2 CPAs with maximum located in $t_1 \approx 5820 \neq t_0$. Regarding the degenerate cases where the correlations are about similar for two key hypotheses, they can be detected by computing the ratio between the first and second best correlations. If it is too close to 1, then both hypotheses can be kept. As these situations happen seldom, very few sboxes candidates will remain after SCARE: a brute force trial of all the candidates will easily discard the wrong hypotheses.

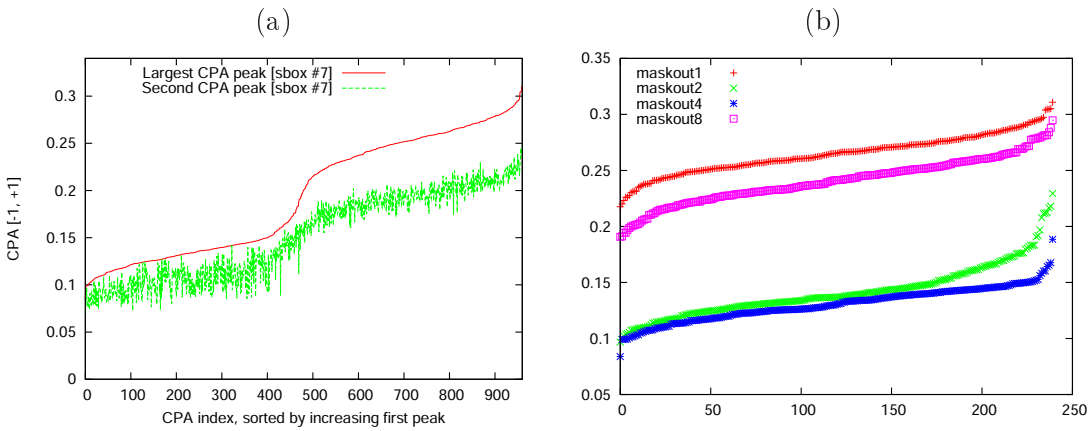


Figure I.11: (a) Correlations obtained for 960 CPAs aiming at retrieving the DES sbox #7; (b) Correlations obtained for each of the 960/4 CPAs aiming at retrieving each fanout bit of the DES sbox #7.

To summarize this section about sboxes retrieval thanks to SCARE, one can say that some adjustments are required for the power analysis to work on one $2 \rightarrow 1$ sub-function, but that subsequent attacks on the $960 - 1$ remaining functions are very reliable. The SCARE oracle is thus a trustworthy tool once tuned.

Appendix J

WDDL is Protected Against Setup Time Violation Attacks

Extended version of article [411]

Authors: *Nidhal Selmane, Shivam Bhasin, Sylvain Guilley, Tarik Graba and Jean-Luc Danger*

Abstract

In order to protect crypto-systems against side channel attacks various countermeasures have been implemented such as dual-rail logic or masking. Faults attacks are a powerful tool to break some implementations of robust cryptographic algorithms such as AES and DES. Various kind of fault attacks scenarios have been published. However, very few publications available in the public literature detail the practical realization of such attacks. In this paper we present the result of a practical fault attack on AES in WDDL and its comparison with its non-protected equivalent. The practical faults on an FPGA running an AES encryptor are realized by under-powering it and further exploited using Piret's attack. The results show that WDDL is protected against setup violation attacks by construction because a faulty bit is replaced by a null bit in the ciphertext. Therefore, the fault leaks no exploitable information. We also give a theoretical model for the above results. Other references have already studied the potential of fault protection of the resynchronizing gates (delay-insensitive). In this paper, we show that non-resynchronizing gates (hence combinatorial DPL such as WDDL) are natively immune to setup time violation attacks.

Keywords: AES; FPGA; Setup violation fault attacks; WDDL; Protection against faults.

J.1 Introduction

Side channel analysis or attacks (SCA) are attacks based on the analysis of the secret information (generally the encryption key) leaked from the physical implementation of the cryptographic system. The leakage is passively observed via timing information, power consumption, electromagnetic radiations, *etc.* Protection against side channel attacks is important because the attacks can be implemented quickly and at a low cost. Differential power analysis (DPA) [248] and its derivatives such as correlation power analysis (CPA) [60] correlate the leakages with an internal power model, which depends on the cryptographic key.

Several countermeasures have been devised to avoid SCA. Dual-rail with precharge logic (DPL) is one of the state-of-the-art countermeasure against SCA. In DPL, the idea is to make the power consumption of the device uniform, thus hiding the crucial information it conceals. Each signal is replaced by true and false representations. *Precharge & Evaluation* phases are alternated to ensure exactly one switching event per cycle. Wave Dynamic Differential Logic (WDDL) [456] is one of the commonly used DPL. Unlike Sense Amplifier based Logic (SABL) [452], WDDL uses standard CMOS cells. Owing to this property, WDDL can be used with any design as no special library is required. Due to the same reason it can be used in FPGAs [458, 459]. It is interesting to note that WDDL is prone to the "early evaluation" vulnerability [439], corrected from instance in SecLib [189, 175]. Despite this second-order issue, WDDL is relatively secure for a reasonable overhead. Hereafter we present our work with respect to WDDL designs.

Differential fault attacks (DFA) [50, 45, 47] also referred to as active attacks alter the functional behavior of the attacked device by injecting one or several faults. Several techniques are available to inject faults: variations of the supply voltage or clock frequency, temperature variation or irradiation by a laser beam which leads to a wrong computation result that can be exploited to perform DFA. Some countermeasures for DFA have been introduced. These countermeasures are generally based on temporal [26] or spatial [288, 236] redundancy, either in a generic manner or taking advantage of some peculiarity of the algorithm.

Here in this paper we analyze the security of WDDL against setup violation fault attacks. We implemented AES (*Singlerail & WDDL*). "Singlerail" refers to simple version of AES, playing the role of the unprotected reference, and "WDDL" is the DPL version. The sbox of the AES is implemented by calculations in composite field GF(2^4) as described in [475].

The results presented in this article are obtained with an EP1S25 Altera Stratix FPGA soldered on a Parallax evaluation board. As described in [412, 242], faults can practically be induced in an FPGA by under-powering the circuit. When we drive the FPGA at a voltage lower than the nominal voltage, the propagation time of the signal increases as illustrated in figure J.1. Such attacks are non-invasive in nature as the attacker does not need access to the silicon die and are therefore easy to implement. We recall that there is no straightforward mechanism to monitor either the power supply level or the frequency in commodity FPGAs. The permanent under-powering causes a

phenomenon called "setup time violation" on one of the timing path of the design causing a faulty byte. We refer to this fault as a "byte-flip" fault, which is obtained by flipping of one or more bits in a byte. The number of bits flipped during a byte-flip is called the Hamming weight of the fault. Since cryptography involves highly complex computations it is very likely that the critical path is in the cryptographic part [123]. Such faults can be exploited using various known attacks [357, 78, 17]. Here we use Piret's attack to exploit the faults and retrieve the secret key using the method described in [357].

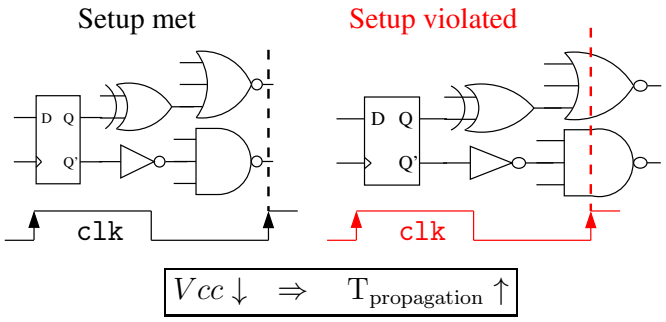


Figure J.1: Setup violation.

The rest of the paper is organized as follows. In section J.2, we explain the WDDL rationale and the design flow to implement it. Section J.3 describes the attack setup and the faults analysis procedure. Section J.4 presents the comparison of a fault acquisition campaign on single-rail and WDDL version of AES, in terms of spatio-temporal localization of faults. Section J.5 is devoted to the theoretical demonstration of the intrinsic immunity of WDDL against setup violation attack on AES. Finally, the section J.6 concludes the paper and opens perspectives for better protecting sensitive cryptographic implementations.

J.2 Wave Dynamic Differential Logic

Power consumption of a standard CMOS cell is dependent on the transition of its input. Thus for a DPA-resistant design, a possible solution could be to introduce a family of DPA resistant cells. In a WDDL cell [456], one transition per cycle is observed, which is favourable for a DPA resistant logic style.

WDDL uses true and false representations of each signal (I/O of each cell). To make the power consumption fairly uncorrelated to the processed data, it is necessary that there should be the same number of transition every cycle. This condition is fulfilled by alternate cycles of precharge and evaluation. In the precharge phase all the signals are charged to the same level (*e.g.* 0 in WDDL) and during evaluation exactly one of the two complementary outputs is evaluated (=1). Figure J.2 shows the timing diagram of WDDL AND gate. We can see that during precharge all signals are put to logic 0.

During evaluation, exactly one of the two complementary inputs and outputs evaluates to 1.

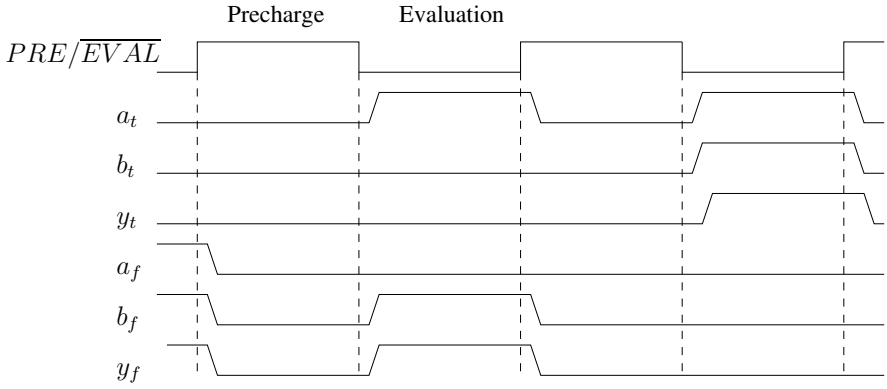


Figure J.2: Timing diagram for a WDDL AND gate.

In DPL, glitches make the design vulnerable to attacks [294]. Indeed, without special attention, if the inputs arrive at different moments, glitches can be observed. To avoid glitches it is necessary that all the gates in the design should be positive in nature. To ensure this in WDDL, the design is synthesized with a library consisting of only positive gates (like AND, OR) [204]. As shown in figure J.3, a WDDL AND gate consists of an AND gate (G) and a complementary OR gate (G^*, satisfying $G^*(x) \doteq \overline{G(\overline{x})}$). For sequential circuits, each flip-flop is replaced by a pair a flip-flops. This double flip-flop allows the precharge wave to propagate through the whole design as all the gates are positive. It has to be noted that inverters in WDDL are implemented by crossing the true and false signals of the same variable.

A point worth noting in figure J.3 is that one flip-flop in the single-rail design is replaced by four flip-flops in the WDDL design. This is explained as follows. During the precharge phase, the combinatorial part of the circuit will be discharged to 0 and this 0 is stored to the first of the two flip-flops. The second flip-flop will store the result of the last computation. In the evaluation phase, the value stored in the second flip-flop serves as input and the output is stored in the first flip-flop. In the mean while, the zero stored in the first flip-flop is shifted to the second flip-flop to allow proper precharge of the circuit ahead in the next cycle. This phenomenon happens in both true and false rail. Thus the number of flip-flops is quadrupled in the WDDL design.

In our implementation, we use a different way to ensure all positive logic. Instead of using positive gates, we use a library containing all look-up tables (LUTs) which implement a positive function. This technique is called WDDL+ in [205].

J.2.1 Design Flow for WDDL Implementation

As every digital system, cryptographic coprocessors can be separated into control and datapath parts. As the secret key is used only in the datapath part, leakage from

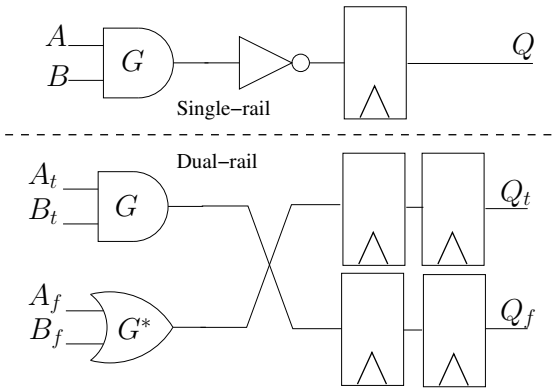


Figure J.3: WDDL building block.

the control part is not crucial. Thus to assure security of the design it is sufficient to implement only the datapath in WDDL. This will also save area as WDDL takes more area on the FPGA than a single-rail design. The design flow to implement a cryptographic coprocessor on an FPGA is shown in figure J.4. The datapath is first synthesized using an ASIC synthesizer taking advantage of a library with only positive LUTs (the FPGA synthesis tool does not provide enough options to limit the library therefore we use an ASIC synthesizer). As the number of positive functions with four inputs is fairly large (166), the library size is reduced by keeping only one function for any equivalence class where the inputs or the output are logically inversed and the inputs are swapped. Indeed, the inversions are dealt with externally from the LUT with wire-crossings (typical transformation of WDDL), and the FPGA mapper tools are able to change the LUT mask to make up for input pins permutations. Then the output netlist is processed using a custom tool (called vDuplicate [185] in figure J.4) which converts a single-rail netlist into a WDDL netlist. The controller is then connected to the WDDL datapath using a wrapper. The FPGA vendor tool does synthesis, mapping, placing & routing for the whole design on the FPGA.

J.2.2 Dualization of single-rail design

As mentioned earlier, the controller of the cryptoprocessor is not converted to WDDL technology. To make the same controller work with the WDDL version of the datapath, there is a need to introduce an extra input to the controller. As the WDDL datapath will precharge in one cycle and evaluate in the next cycle, we require the controller to work every alternate cycle (evaluation) and freeze during the precharge phase. A enable signal driven at half the clock frequency is introduced to provide this functionality.

One more modification is required in the design. The I/Os of the WDDL datapath are dual-rail, while the signals from controller to datapath and the global I/Os are single-rail. Therefore we need to create a wrapper which will make the single-rail and the WDDL parts compatible. As shown in figure J.5, all the inputs to the datapath (I & C)

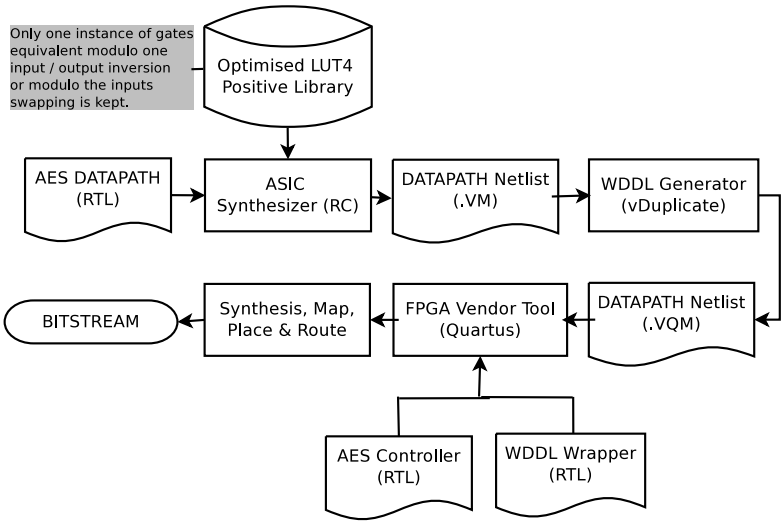


Figure J.4: WDDL design flow.

are transformed into dual-rail (true and false) signals using inverters. A signal phase is introduced to make the datapath inputs compatible with precharge and evaluation phases. When phase is precharge, both the true and false inputs are discharged to 0. During the evaluation phase, exactly one of the complementary input charges to 1. For the output (O) as shown in figure J.5, the true output is ANDed with the inverted false output. Only taking the true output while leaving false output unconnected is also an option. The reason for using both the outputs is to make sure that the FPGA vendor tool doesn't remove the unconnected false output during optimization in placement and routing steps, as the optimization will create an unbalanced design. After the wrapper has integrated the WDDL datapath and the controller, it seems to work as a single-rail design from the top. Therefore now the design could be simulated, synthesized or tested as a single-rail design using the same softwares. In this way, if the results are same for single-rail and WDDL, we are ensured that the two designs are functionally equivalent.

When both the designs are synthesized, the single-rail design works at a frequency of 54.5 MHz using 10% of the FPGA logic. On the other hand, the WDDL design as expected works at a frequency of 22.01 MHz which is less than the half of the single-rail frequency due to two phase operation in WDDL. The WDDL design uses 51% of the logic blocks i.e. 5 times more than the single-rail design. The overhead of 5 times is due to the fact that 20 instantiations of sbox is used and each sbox is replaced by a true and false sbox which makes the design huge.

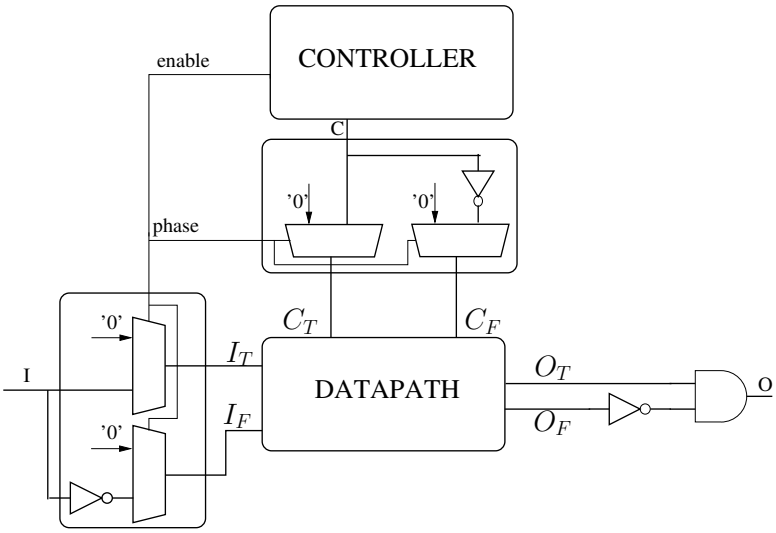


Figure J.5: Basic architecture of WDDL wrapper.

J.3 Setup for fault attacks on FPGAs

In order to induce faults during the execution of the algorithm we drive the core of the FPGA at a non-nominal continuous voltage Vcc. The power supply is remotely controlled using GPIB cable.This feature allows us to test various values of input voltage successively. For each value of Vcc, the triples {message, key, ciphertext} are recorded for 1,000 encryptions at each 100 values of Vcc. Figure J.6 sketches the experimental setup. The testing platform is a stratix FPGA soldered on a parallax board. The FPGA is powered by the programmable power supply. The rest of the board is powered by a 5V constant supply.

Once acquisition is done, we use a software for an off-line analysis of the collected ciphertext in order to detect single byte errors that occur during the encryption. A modified register transfer level (RTL) description of AES where faults can be injected at any byte of the ten rounds is used to generate a dynamic database. The database consists of all possible ciphertexts generated only by single faults for each key, message pair. Then we test if the faulty ciphertext matches an entry in the database. If so, the fault is said “covered” otherwise the fault is said “uncovered” and the ciphertext is affected by multiple byte faults. If a fault is covered, the software provides the location, the value of the faulty byte, its corresponding correct byte and the Hamming weight of the fault. The purpose of this implementation is to identify the typology of single faults that occurs in the FPGA.

If the fault is ”covered” then we can identify the round and the sbox affected by the fault. This concludes that the voltage reduction generates random single faults that most differential attack models are based on. In this experiment we use a global non-invasive

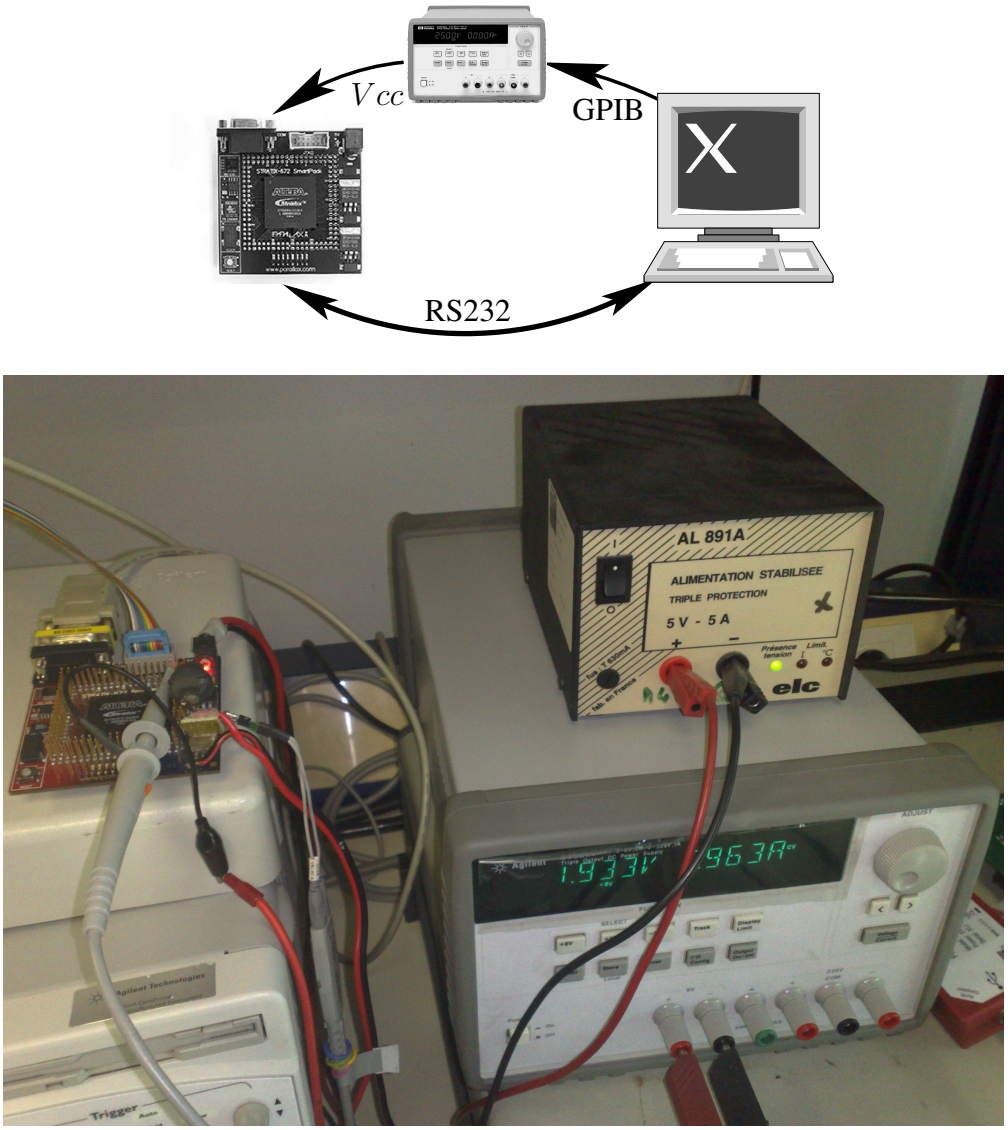


Figure J.6: Experimental platform.

fault injection technique. The propagation time increases with the decrease of the power supply and faults are caused by an early latching of a combinatorial function as shown in figure J.1.

J.4 Experimental Results

In this section, the fault analysis is used to find the occurrence of a single byte fault that affects the state matrix of AES. Both single-rail and WDDL versions are tested

against setup violation faults. Faults detected are those occurring only in the datapath, while the key schedule is assumed here to be fault-free. Indeed, in our design, the key schedule block is not critical in timing.

Figure 7 shows the occurrence of faults in single-rail implementation. We can see that the graph of single faults has a bell-shaped distribution. As we decrease the voltage beyond a certain threshold, setup time is violated on multiple paths and faults become multiple (uncovered). The maximum percentage of single faults is 39% at a voltage of 1.256 V as shown in figure 7. All single faults are analyzed in terms of spatio-temporal locality: Figure 8 and figure 9. 26% of single faults occur in round 8 and 12% of them occur in round 9 (refer figure 8). Such faults are exploitable using Piret's Attack. Thus the single-rail implementation of AES with SBOX in LUT is not protected against "setup violation" attacks.

For the WDDL version of AES, the results are shown in the figure 7. Since we use only positive LUTs to implement WDDL, there are no glitches in the circuit. When we run the fault attack campaign on WDDL design, less than 2% of the detected faults are single and all of them fall in the last round of AES as shown in figure 8. These faults are not exploitable and thus the key cannot be retrieved using Piret attack. The software for fault analysis allows us to see faulted bytes and its corresponding correct value (value of byte if not faulted). We find that everytime a fault occurs, the faulted value C^* is less than its corresponding correct value C, in a bitwise sense: $C \,\&\, C^* = C^*$. This comes down to using the partial order $\preceq$, defined bit by bit in the following truth table:

C	C^*	$C^* \preceq C$
0	0	1
0	1	0
1	0	1
1	1	1

This means that all the faults are caused when an expected '1' takes a value equal to '0'. Tables J.1, J.2 and J.3 show some examples of practical faults.

We have checked that the bytes faulted at value **0x00** are not due to any transmission problem of the ciphertext to the PC through the UART. Indeed, in the design presented in Fig. J.6, the critical path is by far in the AES and not in the UART.

Table J.1: Single fault in round 10.

key	00000000000000000000000000000000
message	093c7b78f4fa44baff2f67fc2d259dd0
ciphertext	96296994aba80db3ea81b491230985db
ciphertext*	96296994aba80db3ea81b49123090 0db

Table J.2: Single fault in round 9.

key	00000000000000000000000000000000
message	c4968c64c72bbcb88acb744253f51be7
ciphertext	43720bee23f577a8311bf769f58e97e7
ciphertext*	00720bee23f57700311b0069f5 00 97e7

Table J.3: Fault strictly before round 9.

key	00000000000000000000000000000000
message	be6d1ddeb2406e9a8546efc65284c4e7
ciphertext	fa73bc0ffb30e9209ec8bfe8f77b96f4
ciphertext*	00000000000000000000000000000000

The reason why all the faults are seen in the last round is as follows. When an XOR gate is implemented using positive logic, it is a combination of AND, OR gates and inverters (for inverted inputs). These inverters yield a mixture of true and false part of the design as per the definition of XOR. Thus a fault occurring in a true part is further corrupted by mixing with the false part and vice versa. The MixColumns operation involves a lot of XOR operations. Therefore a MixColumns operation after a fault will corrupt the fault which cannot be detected. Since the last round does not have MixColumns, the faults are detected but not exploitable. One interesting observation was that every time a byte is affected by a fault, a null byte in the ciphertext was reflected at its expected place. This means that even after successfully injecting the fault during encryption and precisely knowing the location of the fault, the output does not give any information which can be acted upon to retrieve the hidden secrets. The results observed are easily reproducible. This means that for a particular voltage lower than the nominal voltage, if the ciphertext and input message are constant, the fault is often in the same sbox. This feature gives us better flexibility for complete analysis of these faults. Therefore, a WDDL design is naturally secure against setup violation faults. This has been further explained in the section J.5.1.

J.5 Theoretical Fault Analysis

The purpose of this section is to show that the fault model corresponding to a setup violation time has the consequence that all DFAs on AES in WDDL are impractical.

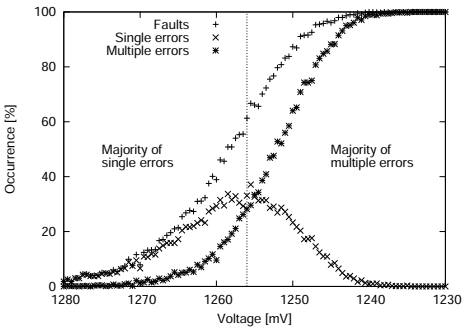


Figure 7: Occurrence of Fault — Singlerail.

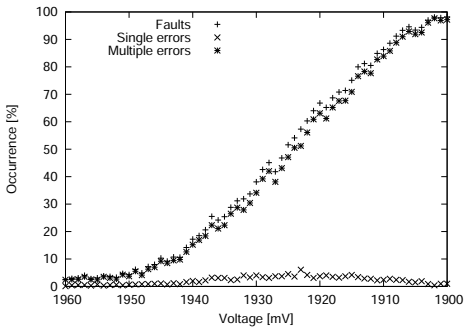


Figure 7: Occurrence of fault — WDDL.

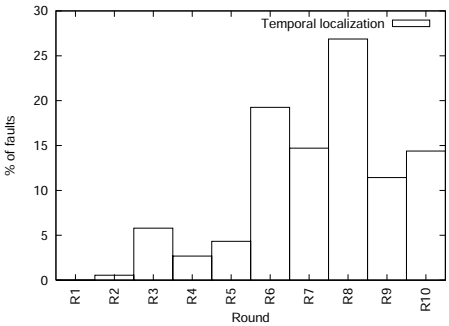


Figure 8: Temporal localisation — Singlerail.

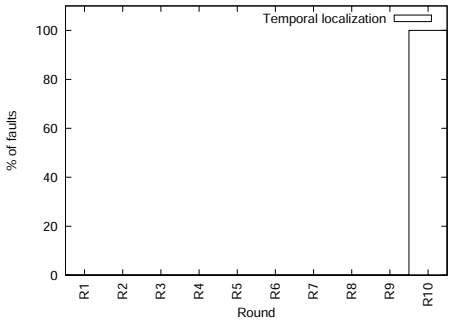


Figure 8: Temporal localization of fault — WDDL.

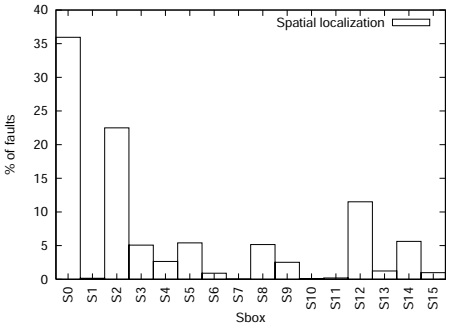


Figure 9: Spatial localisation — Singlerail.

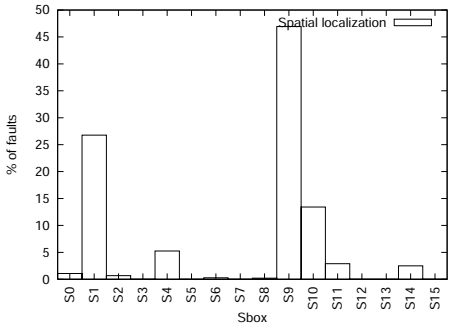


Figure 9: Spatial localization of fault — WDDL.

J.5.1 Fault Analysis on AES in WDDL with SubBytes in LUTs

J.5.1.1 Fault Model

In an under-powering or over-clocking attack, faults arise from a setup time violation [412, 242]. Authors of paper [129] argue that the effect of a glitch on the power supply increases the propagation times of all the signals, which makes this disturbance similar in effect to the global chip under-powering. As the WDDL protocol with a (0, 0) spacer starts in evaluation step with all the nodes voltage equal to zero, the evaluation consists in propagating rising transitions along exactly half of the wires. If by any means, an attacker manages to trigger a setup time violation, the consequence is an asymmetric bit flip: only 1 to 0 errors are considered. Therefore, the consequence of the fault is to leave (at least) one dual-rail signal in its (0, 0) precharge state, while the others couples of wire are in legal (0, 1) or (1, 0) evaluation state.

As already discussed in Sec. J.4, the error is likely to happen for a few dual-rail signals if the stress level is low. This invalid data representation will then propagate through the next round logic. Four cases are possible:

1. the protocol error can turn into functional errors on the data or not, and
2. the protocol errors can vanish while flowing through the combinatorial logic (self protocol healing), or, at the opposite, be amplified.

The next section shows that functional errors occur, corresponding to bits erasure. In addition, the erasure rate increases: one single error at the entrance of a round will trigger many invalid precharge bits to be generated, and we show that in a reasonable cryptographic algorithm (no computation is done uselessly), the erasure rate increases. The consequence is that, after some percolation in the combinatorial logic, most of the values are erased.

J.5.1.2 Propagation of Faults

We start this analysis by the example of two representative gates: the AND and the XOR functions each having two inputs that we note a and b. We assume in this study that the fault occurs on input a. In evaluation, instead of having $(a_t, a_f) = (0, 1)$ when $a = 0$ and $(a_t, a_f) = (1, 0)$ otherwise, we simply have $a_t = a_f = 0$, which can also be expressed as a=NULL. The logic that implements the AND gate is $(c_t, c_f) = (a_t \cdot b_t, a_f + b_f)$. When a is faulty, the Tab. J.4 function degenerates to AND$(a^*, b) = 0$ if $b = 0$, and NULL otherwise.

The same analysis can be carried out for the WDDL XOR gate in figure 10. The logic that implements the WDDL XOR gate is $(c_t, c_f) = (a_t \cdot b_f + a_f \cdot b_t, (a_f + b_t) \cdot (a_t + b_f))$. This equation shows that if we have a faulty input $(a_t = a_f = 0)$ then the output will be NULL $(c_t = c_f = 0)$. Thus the XOR gate has a maximum error propagation since the error is propagated for any value of b as shown in table J.5.

Now, for any function f, we have this property:

Table J.4: Modified functionality of an AND gate in the presence of erasure faults.

Correct computation								
a	b	a_t	a_f	b_t	b_f	c_t	c_f	c
0	0	0	1	0	1	0	1	0
0	1	0	1	1	0	0	1	0
1	0	1	0	0	1	0	1	0
1	1	1	0	1	0	1	0	1
Faulted computation								
a	b	a_t	a_f	b_t	b_f	c_t	c_f	c
NULL	0	0	0	0	1	0	1	0
NULL	1	0	0	1	0	0	0	NULL

Table J.5: Modified functionality of an XOR gate in the presence of erasure faults.

Faulted computation								
a	b	a_t	a_f	b_t	b_f	c_t	c_f	c
NULL	0	0	0	0	1	0	0	NULL
NULL	1	0	0	1	0	0	0	NULL

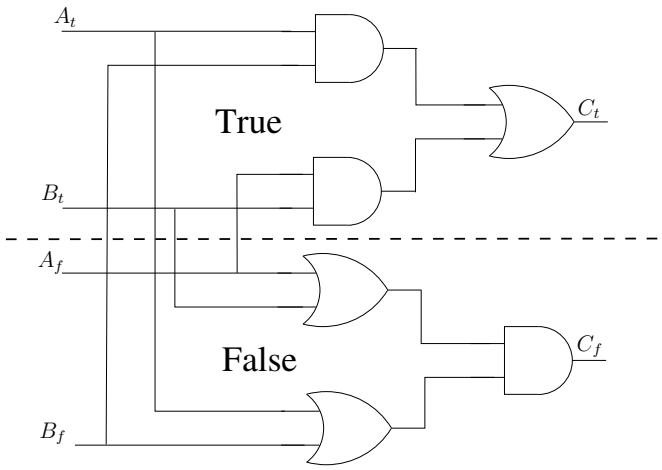


Figure 10: WDDL implementation of the XOR gate.

Definition 1. *Let f be a positive Boolean function with inputs (a, b) then its WDDL equivalent F can be defined as:*

$$\begin{cases} F_t(a_t, b_t) &= f(a_t, b_t)\,, \\ F_f(a_f, b_f) &= \overline{f(\overline{a_f}, \overline{b_f})}\,. \end{cases}$$

The output of f is correct when f does not depend on the faulty input, and erased otherwise.

The proof is straightforward. If the output does not depend on the faulty input, the computation is correct for both the true and the false outputs, because the protocol violation does not impact the result. On the contrary, for the configuration of non-faulty inputs b such as F depends on the faulty input bit, then we have four cases:

1. $F_t = F_f = 1$: impossible since F is positive and the inputs are lower than a legal value, that is either (1, 0) or (0, 1),
2. $F_t = 1$ and $F_f = 0$. In this case, $1 = f(0, b)$ [equation for F_t] and $0 = \overline{f(\overline{0}, \overline{\overline{b}})} = \overline{f(1, b)}$ [equation for F_f] , *i.e.* $1 = f(1, b)$. Therefore $f(0, b) = f(1, b)$. However, we assumed that F does depend on the first faulty input, hence a contradiction.
3. $F_t = 0$ and $F_f = 1$: for the same reason, this case is incompatible with the input configuration such that F does depend on the faulty input.
4. Consequently, the only possibility is that $F_t = F_f = 0$, hence a NULL propagation.

Let us now study a random function, modeling the byte substitution table (SubBytes) of the AES. If there is a NULL fault at the input, then:

- for one half of the input data, a specific output bit will depend on this input, and
- for the other half, the targeted output bit does not depend on the input.

Table J.6: Equations for the bytes transformations $\times 01$, $\times 02$ and $\times 03$.

a'	$a \times 01$	$a \times 02$	$a \times 03$
a'_7	a_7	a_6	$a_7 \oplus a_6$
a'_6	a_6	a_5	$a_6 \oplus a_5$
a'_5	a_5	a_4	$a_5 \oplus a_4$
a'_4	a_4	$a_3 \oplus a_7$	$a_4 \oplus a_3 \oplus a_7$
a'_3	a_3	$a_2 \oplus a_7$	$a_3 \oplus a_2 \oplus a_7$
a'_2	a_2	a_1	$a_2 \oplus a_1$
a'_1	a_1	$a_0 \oplus a_7$	$a_1 \oplus a_0 \oplus a_7$
a'_0	a_0	a_7	$a_0 \oplus a_7$

Therefore, statistically, one half of the output bits are erased to NULL. Notice that this result is independent of the exact functional decomposition in a positive dual gates netlist. Similarly, if two inputs are erased, then 3/4 of the outputs will also be NULL. And of course, when seven or eight errors are presented at the input, all the output bits become NULL.

We have already shown in section J.5.1.2 that with XOR gates the fault propagation is maximal. The MixColumns transformation is a multiplication of a polynomial over $GF(2^8)$ with the fixed polynomial $a(x)$[J.1], reduced modulo $x^4 + 1$.

$$a(x) = (\texttt{0x03})x^3 + (\texttt{0x01})x^2 + (\texttt{0x01})x + (\texttt{0x02}) \tag{J.1}$$

The equations for the byte multiplications involved in this multiplication are written down in Tab. J.6. Hence we see that the MixColumns operation is implemented as a tree of XOR gates. This ensures a maximum propagation of NULL.

In an SPN (substitution permutation network) like AES, the fault number can only grow at each step. Indeed, for every block f, if a fault is stopped, then: $f(\texttt{'U'}, x)$ is certain, for a given input x. Now, this means that $f(\texttt{'0'}, x) = f(\texttt{'1'}, x)$, and this implies that f is not bijective. Therefore, differential attacks become difficult as the attacker observes an erased value, and cannot backtrack from the faulty ciphertext. The best case being when all the output bits are erased and thus no information that can be useful to generate the key is available.

Unlike byte-flips induced by a laser, the setup time violation on WDDL causes no computation to be wrong. Instead, when an input is partially NULL, the logic evaluates the bits that can be correct for sure, but answers NULL if it cannot decide. Therefore, the propagation model is that of `'U'` in VHDL [229]. The logic tries to evaluate bits that would not be wrong if any correct value (`'0'` or `'1'`) were used instead of `'U'`. We recall in Tab. J.7 the extended truth table of the universal gate `AND` over $\{\texttt{'0'}, \texttt{'1'}, \texttt{'U'}\}^2$.

As shown in Fig. 11, the conversion of the dual-rail signals to single-rail turns a NULL into a `'0'`. This circuit makes use of both true and false signal halves, so as to prevent

Table J.7: Truth table for the universal gate `AND`.

AND	'0'	'1'	'U'
'0'	'0'	'0'	'0'
'1'	'0'	'1'	'U'
'U'	'0'	'U'	'U'

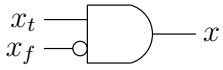


Figure 11: Dual-to-single rail circuitry usable in the case of a NULL0 spacer.

the CAD tool from simplifying half of the logic and balance the true and false networks. Therefore, if a fault occurs during the computation, it can be observed. This difference could be exploited by an attack, as done in the attack of Gilles Piret. However, the computed differential will not disclose any information about the last round key, since the XOR function used to mix it propagates a NULL.

All the considerations detailed regarding WDDL rely on the fact the gates are positive. Indeed, the gates will stick to zero unless valid values are produced. This is not true for delay insensitive gates which stay in a zero state and jam the computation. Notice that in WDDL the results are independent of the type of spacer used. It can be $NULL0 \doteq (0,0)$, $NULL1 \doteq (1,1)$, used as constants or interleaved alternatively or randomly.

J.5.1.3 Generalization to Arbitrary Fault Models

We consider two categories of faults:

1. **Asymmetric faults**, where bits can only be flipped from 1 to 0. This type of faults is typical encountered in WDDL circuits stressed by a global perturbation, such as under-voltage or over-clocking. Glitch attacks can lead to the same symptom, because it manifests in adding a delay globally to all wires. Flash of white light have been reported in [85, §12, page 163] to zero selectively the output of some operations. Equally, laser shots on SRAM-based FPGAs tend to favor $1 \rightarrow 0$ bit-flips over $0 \rightarrow 1$ [286]. Notice that in DPL with a (1, 1) spacer, the opposite transition occurs when trying non-invasive attacks. We do not detail this situation as it is the exact opposite of the 1 to 0 case.
2. **Symmetric faults**, where bits are susceptible of toggling in both directions. Laser shots can trigger both 1 to 0 and 0 to 1 transitions. This fault is thus semi-invasive, as opposed to the previous ones. Therefore, it models a more powerful attacker, at least able to chemically prepare the sample to attack.

In the context of asymmetric faults, DPL circuits are natively protected as such. In this respect, it is interesting to compare the pros and the cons of synchronous and asynchronous circuits. When exposed to under-voltage, asynchronous circuits will continue to work, down to a voltage value where the gates will not be supplied enough to produce a strong one. Below this threshold, errors of type "stuck at zero" will manifest, exactly as in the case of synchronous circuits. Overclocking is not an attack that applies to asynchronous circuits that are, by definition, clockless. However, we have noticed that this perturbation is ineffective is exposing secrets. Therefore, a synchronous circuit will be less reliable in the presence of non-invasive faults, but as secure as an asynchronous circuit. A trade-off between the two approaches can be reached by considering synchronous circuits with jitter on the clock. The jitter can have a large variance, since even if it conducts to a setup time violation, the secrets remain safe. Therefore, with DPL, it is secure when used in addition with aggressive clock jitter.

If the attacker has the means to inject symmetric faults, then three types of protections must be considered:

1. When the fault induction is gentle, single bit flips is the most likely fault model. In this case, even if the fault is a 0 to 1 transition occurring during the evaluation stage, the only risk is to create a (1, 1), also called NULL1. However, in a dual way of the case study of the propagation of NULL0 values, we can show that the propagation of NULL1 consist in an erasure of the data, so that the syndrome does not convey any single bit of information about the faulty circuit internal state. DPL style thus forces the attacker to be less furtive.
2. With a more intense stress, the attacker will start to induce multiple faults with low multiplicity. In this case, a DPL gate can output completely false values. For instance, an AND gate for which the inputs are NULL0 and NULL1 evaluates to the correct value 0 (with respect to WDDL valid states), even if the two unfaulty inputs were both equal to 1. To protect the implementation against those attacks, additional detection hardware must be added so as to cross-check the computation. A little gain can however be obtained: As the DPL style is protected against single faults, a datapath of n bits can be checked with code words of only $n-1$ bits without risking to weaken the security level. A protection method at the technological level such as the one presented in [461, 462] could be extended from SRAM points to DFFs and combinatorial gates. By using high-VT P transistors (those that compute the `'1'`) and low-VT N transistors (those that compute the `'0'`), the designer could make the faults $1 \rightarrow 0$ much more likely than the opposite $0 \rightarrow 1$.
3. When the stress is very strong, then we expect the faults to be very frequent. Hence the recommendation to use physical captors spread on the chip surface.

Now, if we consider only asymmetric faults, we could think that power analysis could be made possible by the fault injection. Indeed, if DFA does not expose the key, it at least indicate to the attacker that a fault has happened. More precisely, we could imagine to correlate the amount of detected faults to a side-channel, in a view to establish correlations. Indeed, in nominal operation conditions, the activity is constant: half of

the gates commute in each clock cycle. When a fault is injected, the activity will become lower:

- in a fault position dependent fashion (for sure), as illustrated in Fig. 12,
- but perhaps also in a data dependent fashion.

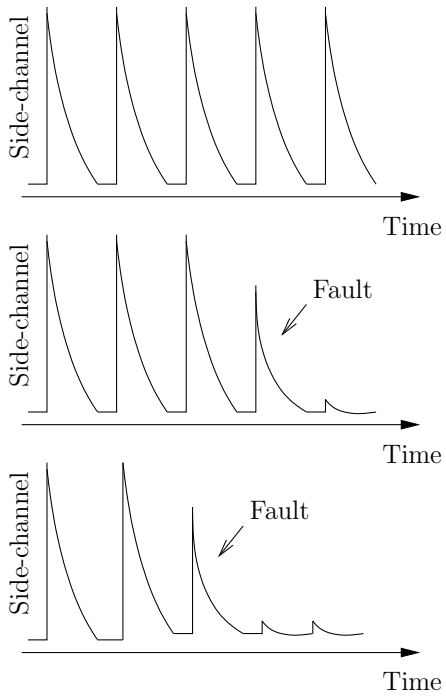


Figure 12: Power dependence of a WDDL circuit in the faults.

However, such an attack cannot be mounted, since if a sensible variable is faulty, irrespectively of its value, the fault will generate a NULL0. Therefore, after the fault, the system has forgotten its value, and computation (in terms of number of toggles) will continue in similar ways. This argument is confirmed by the practical observation of power consumption of WDDL AES as shown in figure 13. We can see that the power consumption of the device is abruptly reduced as soon as the fault occurs approximately at time 2130 ps. The power consumption further reduces after two cycles and remains constant till the end of encryption. It takes exactly 2 cycles (1 ShiftRows and 2 MixColumns) for NULL0 to diffuse through the whole design. This holds even if the DPA protection has a second order flaw, such as early evaluation. The only way to take advantage of such a flaw is to exploit it without faults. Indeed, to rephrase why DFA does not help the DPA, with faults, the distinctions of power curves at second order simply disappear. We cannot show any experimental curve to illustrate this point since we have no mean to deduce the bit concerned with the fault based on the sole knowledge of the ciphertext.

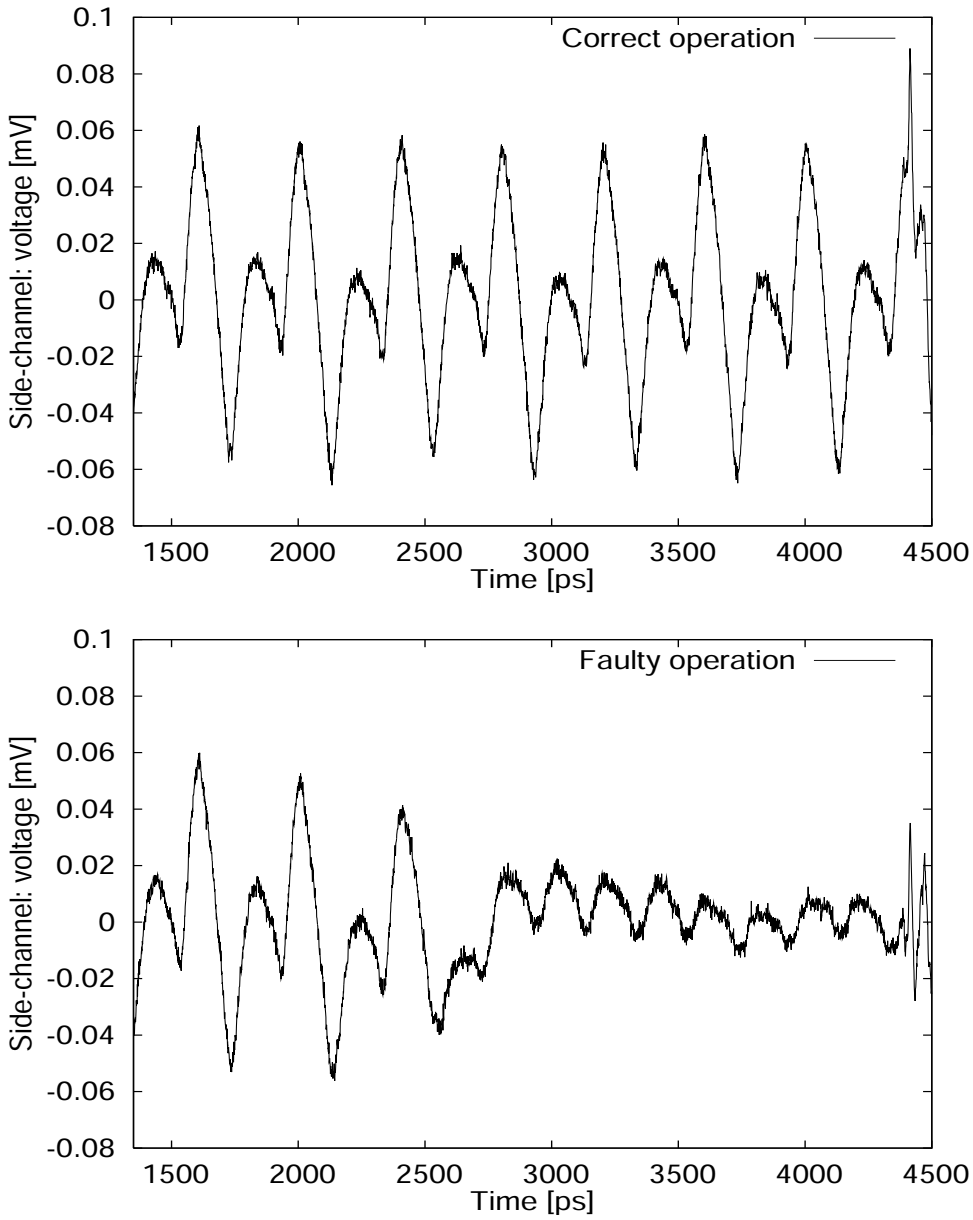


Figure 13: Practical power consumption of a WDDL circuit, without faults (*top*) and in the faults (*bottom*).

Finally, we attract the reader's attention to the fact that vulnerability analysis of WDDL against faults exploitation or DPA in the presence of faults has been argued in the precharge to evaluation step. However, it can be transposed without any change to the case of evaluation to precharge step. Indeed, the circuit's behavior is unchanged,

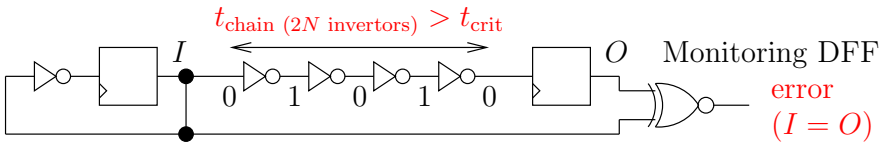


Figure 14: Counter-measure based on the insertion of a monitoring logic with a propagation time larger than the critical path of the rest of the circuit.

except that vulnerable transitions, previously $0 \rightarrow 1$, are replaced with $1 \rightarrow 0$. However the attacker has less insight, since she cannot observe the faults occurring in the precharge stage, that are filtered out by the WDDL circuit wrapper described in Sec. J.2.2.

J.5.2 Counter-Measures against Non-Invasive Attacks

Permanent stress, such as a continuously low voltage or high frequency, generates faults that are trivially undetected by countermeasures based on timing redundancy. Indeed, the same fault is very likely to happen each time the same computation is executed: as the results are consistently false, they are wrongly assumed to be valid. In particular, the double-data rate computation template [288] cannot be used as a protection against the exposure to a steady stress. Thus different countermeasures must be thought of: those based on information redundancy are suitable. We describe below a much cheaper alternative.

A straightforward countermeasure against non-invasive attacks on various circuits (not only DPL) consists in inserting into the circuit some logic in charge of detecting abnormal situations before the critical parts of the designs become faulty. For instance, the figure 14 presents a setup consisting of an even number of inverters, making up a delay line, inserted between two registers. The source register inverts its value every cycle and the combinatorial chain of even number of inverters computes always the same value. At the end of the chain, a destination register checks that the value computed by the chain. The setup time of the invertor chain is violated if the monitored (output) value O is different than the previous input (or equal to the current input I). Hopefully, if the chain is designed to be longer than the critical path, an alarm is raised before the cryptographic parts of the design become faulty.

The chain should be implemented in such a way that it operates at the same clock as the protected circuit and driven by the same source voltage. We implemented this countermeasure on an Altera Stratix FPGA [410, Sec. 5]. Instead of using RTL invertor, we used an "Lcell", the Stratix primitive cell for delay elements, implemented in a LuT resource. This is depicted in Fig. 15. The advantage of using Lcells is that the user is sure that synthesis tool will not remove or shorten the length of the chain during pre- or post-P&R optimization.

We analyzed the chain in order to find a relationship between the length of the chain and the voltage at which the first fault occurs. On the same designs, we also search for the

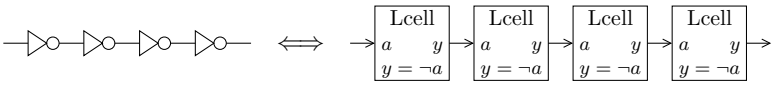


Figure 15: Mapping of the chain of invertors involved in the countermeasure presented in Fig. 14.

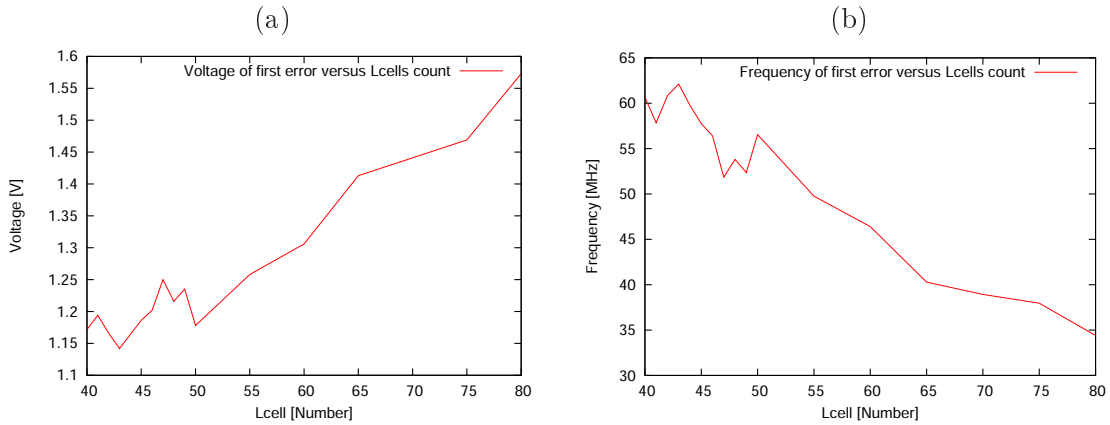


Figure 16: Maximal voltage at nominal frequency (a) and minimal frequency at nominal voltage (b) where the countermeasure of Fig. 14 detects a fault, for 40 to 80 Lcells in the chain.

minimal frequency (at nominal voltage) at which the chain detects a fault. Figure 16(a) shows the voltage of the setup time violation in function of the Lcell number used in the chain. It is clear that the violation voltage increases more or less affinely with the number of buffers. Figure 16(b) shows the same characterization with respect to over-clocking.

The two characterizations done in Fig. 16 allow for a reasoned adjustment of the countermeasure and for the estimation of its tolerance margin. It is also interesting to plot the relationship between the “critical frequency” and the “critical voltage”. This information, plotted in Fig. 17, provides the equivalence of device sensitivity [266] against two global means of injecting faults (over-clocking and under-feeding) [233, Chap. 17].

J.6 Conclusion

Information masking and hiding are two concurrent protection techniques against side-channel attacks. Last year at FDTC’08, Arnaud BOSCHER and Helena HANDSCHUH showed that masking does not protect against fault attacks [52]. On the contrary, we have demonstrated theoretically and shown practically that information hiding (such as DPL) makes it difficult to mount fault attacks, since faulty outputs reveal no information about the keys. Unlike the "differential behavioral attack" (DBA [386]),

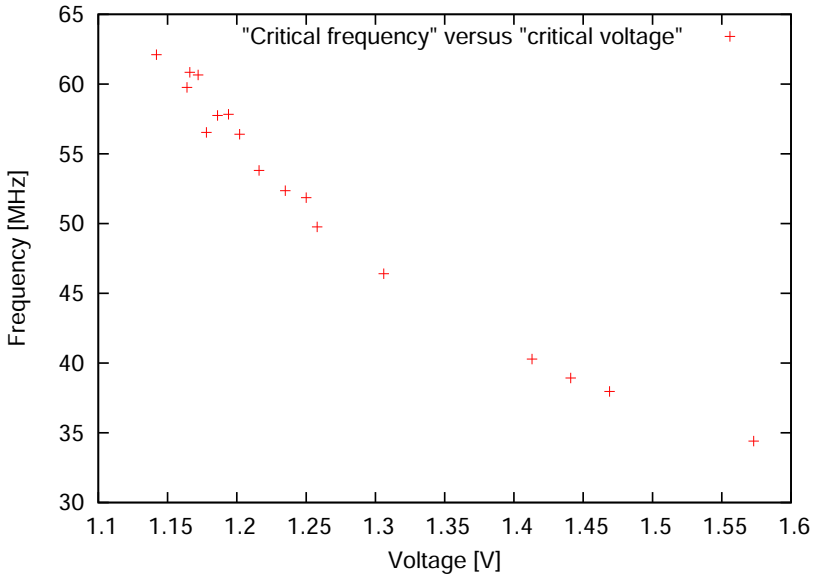


Figure 17: "Critical frequency" *versus* "critical voltage", for various Lcell numbers in the countermeasure of Fig. 14.

where a simultaneous observation of the faulty message and of the power curve empowers an attacker into mounting an attack, in the case of WDDL the attacker cannot learn anything from power curve corresponding to a faulty encryption.

We show, for the first time, that asymmetric fault attacks in general being not a threat for DPL circuits. As a perspective, we can study whether or not more traditional faults model (such as the byte-flip caused by a laser spot) also leads to unsuccessful attacks. Provided this analysis turns out to be correct, all previously proposed countermeasures against DFA for WDDL would be useless: for instance, the alarm (namely the (`'1'`, `'1'`) state) propagation scheme presented in [319] warns of a possible attack against which the circuit is already natively immune.

Acknowledgments

The authors would like to acknowledge the support of French National Research Agency (ANR) for this study, through the SeFPGA `https://sefpga.enst.fr/` grant. Some precious advices also came from the outputs of the MARS ANR project. We acknowledge the suggestions of Laurent SAUVAGE about the possible extension from WDDL to any DPL secure styles; this idea gave rise to the article [33] presented in the next appendix K. We are also thankful to the anonymous reviewer who corrected a flaw in our initial setup-violation detection countermeasure. Finally, we are very grateful to the collaboration with STMicroelectronics AST division (Rousset, France) dealing with hardware crypto-processors security improvements.

Appendix K

Combined SCA and DFA Countermeasures Integrable in a FPGA Design Flow

Extended version of article [33]

Authors: *Shivam Bhasin, Jean-Luc Danger, Florent Flament, Tarik Graba, Sylvain Guilley, Yves Mathieu, Maxime Nassar, Laurent Sauvage and Nidhal Selmane*

Abstract

The main challenge when implementing cryptographic algorithms in hardware is to protect them against attacks that target directly the device. Two strategies are customarily employed by malevolent adversaries: observation and differential perturbation attacks, also called SCA and DFA in the abundant scientific literature on this topic. Numerous research efforts have been carried out to defeat respectively SCA or DFA. However, few publications deal with concomitant protection against both threats. The current consensus is to devise algorithmic countermeasures to DFA and subsequently to synthesize the DFA-protected design thanks to a DPA-resistant CAD flow. In this article, we put to the fore that this approach is the best neither in terms of performance nor of relevance. Notably, the contribution of this paper is to demonstrate that the strongest SCA countermeasure known so far, namely the dual-rail with precharge logic styles that do not evaluate early, happen surprisingly to be almost natively immune to most DFAs. Therefore, unexpected two-in-one solutions against SCA and DFA indeed exist and deserve a closer attention, because they ally simplicity with efficiency. In particular, we illustrate a logic style, called WDDL without early evaluation (WDDL w/o EE), and a design flow that realizes in practice one possible combined DPA and DFA counter-measure especially suited for reconfigurable hardware.

Keywords: Side-Channel Analysis (SCA), Differential Power Analysis (DPA), Dual-rail with Precharge Logic (DPL), Early Evaluation (EE), Differential Fault Analysis (DFA), Wave Dynamic Differential Logic (WDDL), Computer-Aided Design (CAD), Field Programmable Gates Array (FPGA).

K.1 Introduction

Embedded systems that contain cryptographic modules are becoming commonplace with the generalization of privacy, authentication and integrity in digital communications. The cryptographic hardware is very resource consuming because it relies on complex operations needed to prevent illegitimate users from spying, impersonating or altering the communications. Therefore, many studies focus on the optimization of cryptographic blocks. In parallel, new threats – not of cryptanalytic nature – have shown up: it has been suggested and demonstrated that an attacker can break the logical security conveyed by the cryptography by merely observing or perturbing it on the physical layer. The common point between those two exaction strategies is their aim to defeat the security by retrieving some secret elements (such as keys) from which the security features stem.

On the one hand, observation attacks are also known as side-channel attacks (abridged "SCAs" [248]), in that they exploit a physical leakage of the device to gain information about its internal secrets. On the other hand, perturbation attacks consist in altering the state of the device so as to retrieve faulted outputs, that together with nominal outputs, can disclose or negate relationships within the secret bits normally concealed into the hardware; these attacks are referred to as differential fault analyses (abridged "DFAs" [45, 357]). The main strength of SCAs is their furtivity. As they are virtually impossible to detect, an adequate countermeasure must be vigilant each time the cryptographic engine is in use. On the contrary, the first prerequisite for a DFA to be successful is to actually modify the device's state. A detection strategy can thus be enforced to check for the device operations' integrity. However, the careful check of all components of an embedded system is very fastidious and error-prone. In addition, even if any sensitive data is carefully monitored for integrity, the faults coverage remains an issue. Indeed, if detecting one single error (of unitary bit entropy) is easy using simple parity codes, the detection of multiple errors is more difficult to address. In general, the detection logic complexity is growing exponentially with the faults multiplicity, which quickly becomes deterrent in terms of overhead in practical applications.

One device can be claimed tamper-resistant only if it is protected, at least to some extent, against both SCA and DFA simultaneously. It must be noticed that the efforts to deploy in protection depend on the threat. To be successful, the best attacks known so far require to garner some thousands [1] of side-channel traces recording (in SCAs) [248] but only a couple of faults (in DFAs) [45, 357] from an unprotected device. As a consequence, the need for protection is more stringent against DFA than it is against SCA. This asymmetry is one reason for which the countermeasures against DFA and SCA are nowadays

1. And sometimes only a few hundreds can be enough, as exemplified in the DPA contest [445].

studied separately: this partitioning makes it possible for a designer team to tune the countermeasure efficiency according of the threat urgency, while keeping the flexibility to combine them at the final stage of integration. Another reason why countermeasures against DFA and SCA are considered independently is linked with our state-of-art in defense. The protection against DFA is naturally achieved at an algorithmic level, with the introduction of redundancy in data representation and processing. However, the effective protection against SCA is more subtle, since it requires the removal for any source of leakage through physical side-channels. Therefore, the widespread methodology consists in using dedicated logic gates along with *ad hoc* backend steps. As we know how to resist against DFA before the logic synthesis and to resist against DPA after synthesis, it is implicitly considered obvious that the protection against DFA and DPA should be built one on top of each other.

In this article, we advocate that this methodology is neither natural nor efficient. Basically, we show that a class of strong countermeasures against SCA, namely all variants of dual-rail with precharge logic (DPL) styles which do not suffer from early evaluation (EE), are already protected against the state-of-the-art fault injection techniques. Thus, by subsuming the individual issues of securization against SCA and DFA into a unique problem, we arrive to an original solution that is economic in resources because of its duality w.r.t. both the SCA and the DFA threats[2]. In addition, we show that the countermeasure is all the more efficient as the faults multiplicity is high, which is a property out of reach of traditional protections based on coding theory.

Some previous works have already attempted to provide joint countermeasures against SCA and DFA, but thanks to specific features of FPGAs. For instance, the twain papers [71, 306] show how to resist SCA and DFA when dynamic partial reconfiguration is available. In our article, we achieve the same result even on low-cost FPGAs that cannot be reconfigured at run-time. Also, the conference paper [297] employs the DPL strategy, but with a gate-level integrity check (that resembles [183]). We prove in this paper that this systematic verification is overkill.

The rest of the article is organized as follows. Section K.2 presents the DPL protection against SCA, and motivates for the preference of DPL without EE. In section K.3, the protection potential of DPL (w/ or w/o EE) against DFA is explained. The section K.4 presents a methodology for mapping this protection into FPGAs, and details its performances in terms of resources usage. Finally, conclusions are discussed in section K.5.

K.2 Dual-rail with Precharge Logic Styles against SCAs

The goal of a protection against SCAs is to prevent any attacker from retrieving any information from any internal bit. Various solutions have been proposed to address this requirement. Side-channel *masking* consists in making the activity of sensitive bits random by rewriting the algorithm in such a way that those variables depend on a

2. This approach counters in particular the shrewd threat of "*Passive & Active Power Attacks*", *aka* PACA [10].

external entropy source. Side-channel *hiding* adds redundant logic so as to end up with a constant activity when sensitive bits are manipulated. Each solution has its own pros and cons; some logic styles, based on "masked DPL gates", even mix the two for an improved security. Still, the comparison between these securization options is beyond the scope of this article.

In this article, we focus on the hiding styles. Indeed, as will be made clear in Sec. K.3, those styles combine harmoniously with DFA protection, whereas masking styles do not, as demonstrated in [52]. Information hiding at the bit level can be achieved by a large variety of *ad hoc* encodings and protocols. However, the most convenient ones rely on a so-called dual-rail with precharge representation. Every bit a involved in the algorithm is actually mapped into a couple of wires, named $(a_\mathrm{F}, a_\mathrm{T})$, and called the 'false' and 'true' halves of the dual-rail variable a. The couple $(a_\mathrm{T}, a_\mathrm{F})$ alternates between two values:

1. $(0,0)$ or $(1,1)$, called NULL0 or NULL1, and designated as a NULL token, playing the role of spacer, and
2. $(1,0)$ or $(0,1)$, called VALID0 or VALID1, and designated as a VALID token, carrying the value of a.

One DPL computation alternates NULL and VALID tokens, with the remarkable property that exactly one bit toggle occurs in each transition. A pair of gates $(f_\mathrm{F}, f_\mathrm{T})$ respects the DPL convention if:

- It propagates the NULL values, *i.e.*, if all the inputs are NULL, then $(f_\mathrm{F}, f_\mathrm{T})$ is also NULL.
- It propagates the VALID values, *i.e.*, if all the inputs are VALID, then $(f_\mathrm{F}, f_\mathrm{T})$ is also VALID.

Wave dynamic differential logic (WDDL [456]) has been the first logic style to implement these conditions. WDDL has the nice property to be separable, meaning that f_F (*resp.* f_T) depends only on the false (*resp.* the true) inputs half. However, some other properties have been added afterwards to ensure a secure operation of WDDL. First of all, it has been noticed that on the way from all NULL to all VALID values, glitches could occur if the functions $(f_\mathrm{F}, f_\mathrm{T})$ were not positive [204]. Afterwards, many authors notice concomitantly that the evaluation time depends on the inputs values [253, 390]. An up-to-date list of known DPLs styles used as side-channel information hiding countermeasure is given in Tab. K.1.

The salient features of these logic styles are briefly described below:

- WDDL is the less complex DPL style because it is separable, which makes it possible to reduce the overhead of each dual network.
- MDPL adds some logic on top of WDDL to swap randomly the logic interconnect pairs, in a view to balance the routing mismatches. Indeed, this problem is not addressed directly by WDDL but is left to the layouter [457, 191].
- iMDPL fixes the leakage conveyed by data-dependant evaluation and precharge dates in WDDL and MDPL.
- DRSL combines masking and early evaluation protection, and is optimized to be compact using one standard ASIC cell (`OAI222`) and all RSL [441, 442] gates.

Table K.1: Security features of classical DPL styles.

DPL style + reference	∃ Random?	∃ EE?	Target
WDDL [456]	No	Yes	ASIC and FPGA
MDPL [359]	Yes	Yes	ASIC and FPGA
iMDPL [358]	Yes	No	ASIC and FPGA
DRSL [79]	Yes	No	ASIC
STTL [417]	No	No	ASIC and FPGA
SecLib [193, 189]	No	No	ASIC
WDDL w/o EE [this article]	No	No	FPGA

- STTL is a non-masked improvement of WDDL style free of early evaluation. STTL is however not balanced in structure, as WDDL, and is limited in speed by the slow validation path, by design longer than the path of the data signal pairs. This limitation seriously impedes the throughput of STTL. Eventually, we underline that STTL requires the routing of three wires per logical signal.
- SecLib is non-masked computation style that fixes the EE issue and features a balanced structure. To be exhaustive, we should also mention the NCL (Null Convention Logic) that is a generalization of SecLib albeit deprived from any "structural" balance effort.
- WDDL w/o EE is a logic style dedicated to FPGAs that removes the EE without computing a rendezvous. Instead, each functional half gate receives the true and false inputs, and decides to output the VALID value only when all the inputs are VALID. This behavior can be achieved by a purely combinatorial gate, as depicted in Tab. K.2. The detailed rationale behind the "WDDL w/o EE" style is the following:
  - The gate outputs NULL{0,1} when the inputs are NULL{0,1} or transitional from this value.
  - The gate outputs VALID only when all the inputs are VALID.
  - In case of inconsistent values w.r.t. the DPL convention, the gate outputs an arbitrary NULL value.

  This logic does not evaluate early by design, and propagates errors: if any input is stuck to NULL or if the input is out of specifications, then the output always remains to NULL too. In addition, this logic **does not generate glitches** even if the functionality is not positive, and **can be inverting**. Therefore, the synthesis is more optimized than for plain WDDL.

Table K.2: Look-up-Table (LuT) masks encoding for 4-input LuTs implementing the AND function in WDDL w/o early evaluation.

				AND_T		AND_F		**Input state in the DPL protocol**
a_T	a_F	b_T	b_F	FC80		FAE0		
0	0	0	0	0		0		All NULL0
0	0	0	1	0		0		Transitional from NULL0
0	0	1	0	0	0	0	0	Transitional from NULL0
0	0	1	1	0		0		Faulty
0	1	0	0	0		0		Transitional from NULL0
0	1	0	1	0		1		All VALID: $(a, b) = (0, 0)$
0	1	1	0	0	8	1	E	All VALID: $(a, b) = (0, 1)$
0	1	1	1	1		1		Transitional from NULL1
1	0	0	0	0		0		Transitional from NULL0
1	0	0	1	0		1		All VALID: $(a, b) = (1, 0)$
1	0	1	0	1	C	0	A	All VALID: $(a, b) = (1, 1)$
1	0	1	1	1		1		Transitional from NULL1
1	1	0	0	1		1		Faulty
1	1	0	1	1		1		Transitional from NULL1
1	1	1	0	1	F	1	F	Transitional from NULL1
1	1	1	1	1		1		All NULL1

K.3 Potential of DPL w/o EE for Protection against DFAs

K.3.1 Fault Model

We assume in the sequel that multiple faults can be generated locally (by means of a laser or an electromagnetic injection [369]), but decorrelated one from each other.

K.3.2 Early Evaluation Prevention and Faults Transformations

This article is based on [411], that has already shown that WDDL is immune against multiple asymmetric faults such as those caused by setup violations. Basically, the idea is that asymmetric faults are able to turn any VALID token into a given NULL one. For instance, the fault can induce a mutation from any VALID to the NULL0 spacer. The NULL token can propagate until the outputs, being even amplified. However, the NULL wave propagation acts as an eraser, which means that the outputs have eventually lost any information about the faulted values. A parallel is done in [411] between asymmetrical faults and the logical propagation of `'U'` value in the 9-valued type `std_ulogic` of VHDL (IEEE standard number 1076).

We add in this paper that all dual-rail with precharge logics (DPLs) are actually protected against setup violation attacks. Indeed, they never disclose the faulty result in the presence of a setup violation. Instead, they have two different kinds of behavior:

1. WDDL and MDPL compute results given the inputs, and propagate NULL spacers for the outputs whose values are non decidable. This is the logic behavior of `'U'` in VHDL. One could say that faults in these logics are recessive w.r.t. VALID values.
2. iMDPL, DRSL, STTL, SecLib and WDDL w/o EE propagate the NULL on the fault fanout, even if a VALID value could have been deduced. This is the logic behavior of `'X'` in VHDL. Along with the former phenotypic metaphor, faults in this second class of logics are dominant, or rather contaminating, as their propagation is indeed an unexpected avalanche effect.

The implication is that DPL in itself does not provide a good protection against symmetrical faults. As a matter of fact, it can filter out a NULL (see Fig. 1(a)) and generate a faulted VALID from NULL tokens (see Fig. 1(b)). In contrast, the DPL styles that are EE-free propagate the NULL unconditionally; this feature is even part and parcel of the WDDL w/o EE specification. Additionally, the NULL (behaving like an `'X'`) always absorbs other VALID faults, as shown in Tab. 2.

K.3.3 Propagation of NULL Values Through Substitution Boxes

The fault propagation in logics with EE is exploding in substitution boxes (sboxes). The average number of NULL tokens at the output of various sboxes when one or several NULL tokens of the same type (either NULL0 or NULL1) are at the input has been computed in Tab. K.3 for any logic style subject to EE, such as WDDL or MDPL.

In DPL w/o EE, the propagation is also independent on the implementation. It is also more straightforward as it does not depend on the data: the propagation through

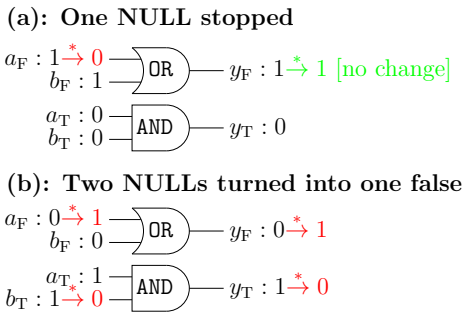


Figure 1: Two DPL w/ EE drawbacks to fight DFAs, illustrated on the example of a WDDL AND gate. In this figure and in the subsequent ones, the asterisk character (*) symbolizes the faults.

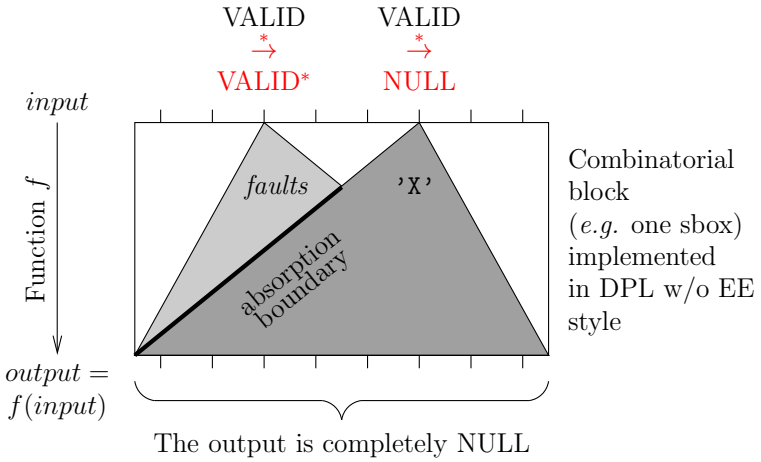


Figure 2: Illustration of the absorption of VALID faults by a salvo of NULL tokens in two interpenetrating logic cones in a DPL w/o EE netlist.

Table K.3: Number of NULL tokens propagated on average through the sboxes of AES ($8 \rightarrow 8$) and DES ($6 \rightarrow 4$) in DPL with EE.

Fault	**AES Sbox**	**DES Sboxes**							
multiplicity	**(SubBytes)**	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	4.04	2.48	2.53	2.65	2.46	2.53	2.60	2.63	2.50
2	7.04	3.88	3.90	3.92	3.93	3.91	3.93	3.93	3.91
3	7.94	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
4	8.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
5	8.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
6	8.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
7	8.00	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
8	8.00	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.

a gate occurs iff the output depends on the given input. This is case of all non-trivial gates. Notably, any fault, even single, on the input of an sbox, corrupts the entire sbox output: the propagation is maximal.

K.3.4 Analysis of the DFA Protection of the Proposed Logic

Single bit faults are inefficient against DPL because they turn a VALID data into a NULL token, that propagates and leads to an unexploitable error since it hides the faulted value. This is the typical scenario described in paper [411]. Highly multiple faults generate randomly a large quantity of NULL values along with some more unlikely but devastating bit-flips. However, as NULL values are systematically propagated, they proliferate very quickly after some combinatorial logic layers traversal. And as they have the nice property to contaminate VALID values, the risky coherent bit-flips (simultaneous $0 \overset{*}{\rightarrow} 1$ and $1 \overset{*}{\rightarrow} 0$ in one dual-rail couple), they jam their propagation hopefully before they reach the algorithm output. This absorption property is all the more efficient as the number of NULL generated by the multiple faults is high. Therefore, the only way to inject a poisonous fault is to stress the circuit sufficiently enough to have multiple faults, without nonetheless creating too many faults so as to leave a chance for them not to be absorbed during their percolation towards the outputs. But, hopefully, in this opportunity window of low stress (generation of 2, 3, or maximum 4 errors because of the high diffusion of cryptographic algorithms), efficient coding schemes can be used in supplement to the DPL w/o EE protection.

To be more accurate, we present a simple model that provides a convincing proof of our assertion. Let us consider a dual-rail circuit that is attacked with a perturbation that is focalized on $2n$ wires, and that has an intensity sufficient enough to cause $m \leq 2n$

simultaneous faults. We also make the optimistic hypothesis that the m faults are equi-distributed over the $2n$ wires, and that the flips are truly symmetrical, *i.e.* it is as likely to flip to a 0 and to a 1. Those conditions model a worst case from the defense view point, because they foster coherent bit-flips susceptible to turn a VALID value into a VALID* one, by the mean of two antinomic flips on two wires pertaining to the same dual-rail couple. To further simplify the modelization, we also assume that the attacked block has a perfect diffusion: in practice, this is not exactly true for one round of an algorithm, but for at least two of them (and exactly two in the case of AES). Nevertheless, it helps us grasp more intuitively the idea of the proof without introducing overcomplicated considerations. Therefore, for a fault to successfully propagate through the round, no single NULL shall be generated. Otherwise, the NULL wave catches the fault, because of the perfect diffusion, as already depicted in Fig. 2. The first noting is that for VALID faults to be generated, m must be even. Indeed, they are generated by pairs. If, on the contrary, m is odd, then at least one NULL (bit-flip of one wire in a pair) is generated, leading to the VALID fault absorption. Then, a VALID fault is generated iff, given a unique fault, a second one occurs in the paired wire. For $m = 2$ faults, this happens with probability $1/(2n-1)$. For more faults, the generation of solely paired faults consists in always pairing the remaining faults. Then, the probability to generate at least one VALID fault that survives until the output is equal to:

$$p(2n, m) \doteq \begin{cases} \binom{n}{m/2} \Big/ \binom{2n}{m} & \text{if } m \text{ is even,} \\ 0 & \text{otherwise.} \end{cases}$$

This probability becomes very small starting from a multiplicity of 4 when m increases up to n [3]. This is to be contrasted with schemes involving a coding with error detection. They are basically able to detect:

- all the faults of multiplicity smaller than the error detection capability r [4], but
- only a ratio of $1 - 1/2^r$ faults for $m > r$.

The figure 3 compares the rate of successful faults injection depending on the multiplicity, for an $n = 8$ set of wires, respectively for the proposed scheme based on DPL w/o EE and for a classical integrity check with a linear code detecting $r = 2$ bits of error.

The authors would like to insist that this is the first time that a countermeasure against DFA proves efficient even in the context of a large number of faults. As a matter of fact, usual schemes, based on spatio-temporal or coding, can be defeated with high probability if the number of faults is greater than the detection capacity. Smartly enough, the implementations using DPL w/o EE take advantage of three properties that all contribute to destroy the VALID faults:

1. faults are very likely to alter only one wire in a pair, especially if the stress is badly localized, thus creating much more NULL tokens than wrong VALID pairs,

3. When m is too large, starting from n, the probability increases, because of the property: $p(2n, m) = p(2n, 2n - m)$.
4. Faults of multiplicity $m \leq r$ mutate a code word into a non-code word.

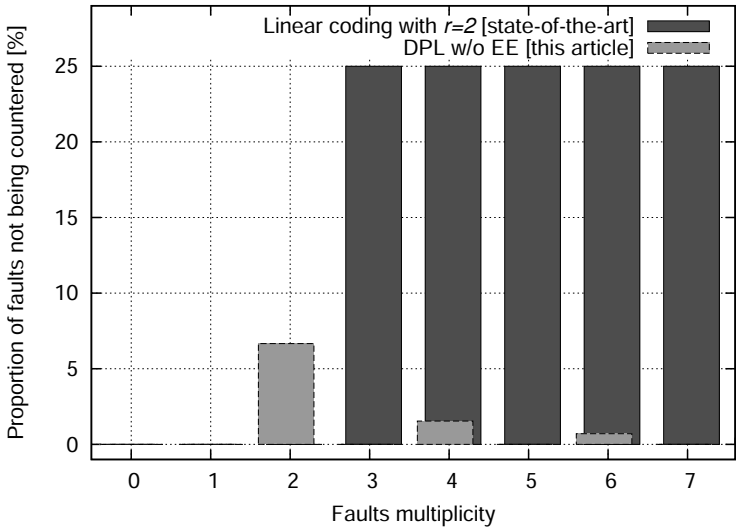


Figure 3: Probability that m faults injected on n wires be innocuous due to the protection conveyed by two different countermeasures: either a *detection* by an informational redundancy scheme or an *annihilation of the faulted data* by one or several VALID $\xrightarrow{*}$ NULL token transformations.

2. because of the protection against EE, NULL values win against VALID ones, hereby hiding in particular VALID fault propagation,
3. as the algorithms implement cryptography, they have a high diffusion, which helps the NULL values meet (and thus eat) the possibly faulted VALID values still alive.

K.4 CAD Flow for the Proposed Counter-Measure

As every digital system, cryptographic coprocessors can be separated into control and datapath. The datapath contains the secret key related operations. Thus to assure security of the design it is sufficient to secure the datapath only. A design flow to implement a cryptographic coprocessor on an FPGA is shown in Fig. 4. Since DPL designs are redundant by nature, we have to use customised tool for processing. The goal of this synthesis is to remove the unnecessary logic redundancy while keeping the redundancy needed for DPL style. This cannot be achieved by a standard design flow. An ASIC synthesizer is used to synthesize the design with a library containing only those gates which respect the DPL style constraints. Then the output netlist is processed using a custom tool which converts a single-rail netlist into a DPL netlist. The controller is then connected to the datapath using a wrapper. Thereafter, a legacy FPGA vendor tool does synthesis, mapping, placing & routing for the whole design on the FPGA. Although the design flow is shown for Altera FPGAs, it has also been tested apt for Xilinx FPGAs.

As stated earlier, to secure a design against SCA and DFA we can use a DPL style

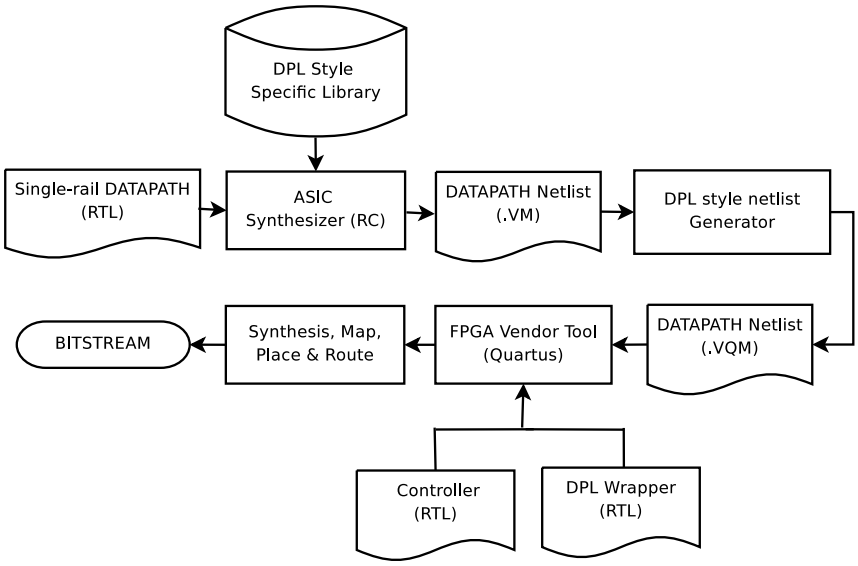


Figure 4: Design-flow for proposed counter-measure.

which is free from EE. WDDL is a DPL style most suited for FPGA designs but it is prone to EE. In [411], authors implement a WDDL design in FPGAs using a library containing four-input functions which are positive in nature. We use the same methodology in this paper. To make WDDL protected against EE, we limit the library to two-input gates, implemented as per Tab. K.2.

We have applied these syntheses on an AES [337] datapath in the Stratix family of Altera. More precisely, we used an EP1S25B672C7 device. The table K.4 summarizes the area of an unprotected datapath, the same datapath protected with an EE-prone logic (namely WDDL) and with an EE-free logic (namely WDDL w/o EE). Both protected designs are embedded in EveSoC [224], and run at similar maximal frequency (27.24 *vs* 27.36 MHz).

The implementation size of the "WDDL w/o EE" style is only slightly greater than that of the original "WDDL", however it is at the same time more secure against SCAs and completely secure against any type of DFAs. The reason why WDDL w/o EE is only 13% larger than WDDL w/ EE is that AES is comprised of many XOR gates, that require the same number of LuTs in WDDL w/o & w/ EE. Finally, we emphasize that the traditional way of protecting a WDDL circuit against faults would have been to use some sort of redundancy (for instance with detection codes), that would for sure represent an overhead in area similar or greater than 13%. This definitely demonstrates that protecting at backend-level (against SCA) a logical detection mechanism (against DFA) is not as efficient in terms of surface as the sole usage of a DPL w/o EE style for both SCA and DFA resistance.

Table K.4: Area of an AES datapath synthesized for the Stratix FPGA.

Logic style	**Reference**	**WDDL w/ EE**	**WDDL w/o EE**
LuT4 count	2,396	12,530	14,126

K.5 Conclusion

This paper shows that, in addition to increasing the resistance against SCAs, the DPL styles also help resist against DFAs. Indeed, single faults consist in turning a VALID token into a NULL one, which conceals the value of the (sensible) data before corruption. The DPL styles that protect against the EE side-channel analysis ensure in addition that the NULL propagation contaminates all the data it crosses in the combinatorial logic cones. Thus, in the case of multiple faults, both VALID faults and NULL tokens are generated, but the NULL tokens destroy the VALID faults prior they arrive at the algorithm observable outputs. Therefore, we show for the first time that a SCA counter-measure is, as such, already an excellent counter-measure against DFA.

We also introduce WDDL w/o EE, a simple logic style that enhances the plain WDDL style by making it EE-free and having it avoid non-VALID tokens propagation. In addition, the synthesis of WDDL w/o EE is efficient because even non-inverting and non-positive functions are allowed. We provide a mapping of this new logic into LuT4-based FPGAs.

Appendix L

Fault Injection Resilience

Extended version of article [206]

Authors: *Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger and Nidhal Selmane*

Abstract

Fault injections constitute a major threat to the security of embedded systems. Errors occurring in the cryptographic algorithms have been shown to be extremely dangerous, since powerful attacks can exploit few of them to recover the full secrets. Most of the resistance techniques to perturbation attacks have relied so far on the detection of faults. We present in this paper another strategy, based on the resilience against fault attacks. The core idea is to allow an erroneous result to be outputted, but with the assurance that this faulty information conveys no information about the secrets concealed in the chip. We first underline the benefits of FIR: false positive are never raised, secrets are not erased uselessly in case of uncompromising faults injections, which increases the card lifespan if the fault is natural and not malevolent, and FIR enables a high potential of resistance even in the context of multiple faults. Then we illustrate two families of fault injection resilience (FIR) schemes suitable for symmetric encryption. The first family is a protocol-level scheme that can be formally proved resilient. The second family mobilizes a special logic-level architecture of the cryptographic module. We notably detail how a countermeasure of this later family, namely dual-rail with precharge logic style, can both protect both against active and passive attacks, thereby bringing a combined global protection of the device. The cost of this logic is evaluated as lower than detection schemes. Finally, we also give some ideas about the modalities of adjunction of FIR to some certification schemes.

Keywords: Fault Injection Attack (FIA), symmetric block encryption, Denial of

Service (DoS), Fault Injection Resilience (FIR), Differential Fault Analysis (DFA), Side-Channel Attack (SCA), Dual-rail with Precharge Logic (DPL).

L.1 Introduction

Secure embedded systems such as smartcards must be tamper-resistant so as to defeat attacks that target directly their implementation. Three kinds of threats have been identified on these devices: perturbation, observation and manipulation. Perturbation attacks consist in covertly changing one data so as to either modify the chip's execution flow or force it to output incorrect results. Observation attacks specifically target the parts of the design that manipulate secrets; their goal is to exploit unintentional side-channel leakages so as to recover sensitive information. Manipulation is an invasive attack that gives to the attacker the power of modifying the chip's functionality or of directly probing signals [216, 133].

Manipulation attacks are the most difficult to resist against, because of their intrusiveness: the device, expected to conceal data, is suddenly reduced into a whitebox system. Fortunately, manipulation attacks involve expensive laboratory equipments, trained personnel and the sacrifice of many samples during their preparation [24]. They are therefore not the most common ones. In addition, efficient countermeasures exist, such as tamper-proof modules (*e.g.* SISHELL and ACSIP solutions by former industrial Axalto) or active shield on top of the chip.

Observation attacks are less costly attacks, since some side-channels, such as the magnetic field, can be recorded at will without the chip even noticing it, in a non-invasive or semi-invasive manner. There also exists a wealth of counter-measures of different quality to make side-channel attacks (SCA) difficult.

Perturbation attacks require a means to alter the device's behavior, without triggering the purported countermeasures that continuously monitor the environment. Some low cost global fault injection attacks (such as overclocking [12, 130, 4], power underfeeding [412, 19, 20] or heating [149, 377, 467]) can be used against weakly protected devices. Most expensive attacks rely on a local perturbation: for instance, laser or particle shots can avoid active shields and thus manage to surgically modify data in extremely well localized zones / dates. At the opposite, those tools can also be used to cause random and extremely spread faults in space / time. With little chance, those highly multiple faults remain undetected and thus successfully alter the chip's state.

Observation attacks on cryptographic blocks usually require a couple of hundreds or thousands observations in absence of countermeasures. At the opposite, fault injection attacks can reveal the secret with a small number of measurements. For instance, RSA [385] computed with the Chinese Remainder Theorem (CRT) can be broken with as few as one faulty computation [49]. The last 128 bit of the key schedule of an AES [337] block cipher can be retrieved with one single well-behaved faulty encryption [463]. These exploits motivate a special focus on fault attacks. This is all the more true as theoretically sound countermeasures have been proposed for SCAs [68] but that the coverage of fault attacks is lacunar: multiple faults, either spread in space or in time, are extremely

difficult to withstand with the state-of-the-art countermeasures. We therefore focus on those attacks in the rest of this article.

Fault injections attacks (FIA) can basically attempt to deviate a targeted device from its nominal functionality in two ways. Either the fault can directly profit to the attacker, such as allowing her to access unauthorized pieces of information, or the fault induces a corrupted computation that the attacker post-processes to recover secrets. The first case is an attack against security mechanisms, whereas the second one targets typically the cryptographic modules. We will not cover the first case, since known methods already exist to cross-check that a punctual valid bit is indeed correct. The second case is at the heart of this paper. Indeed, checking for the correctness of all the steps of a lengthy cryptographic computation is more costly. And above all, we notice that a cryptographic system can indeed remain secure even if it outputs incorrect results. We promote in this paper the idea that, in most cryptographic protocols, it suffices to make sure the fault does not depend on any secret to maintain a provable security level. We call this protection strategy "fault injection resilience", notion abridged as "FIR".

The rest of the paper is organized as follows. The benefit of the FIR over other techniques based on detection is discussed in Sec. L.2. In Sec. L.3, some suitable techniques to implement FIR are described. A case study of a register transfer level (RTL) implementation of FIR is detailed in Sec. L.4. The impact of FIR in two security certification schemes is studied in Sec. L.5. Finally, conclusions and perspectives are given in Sec. L.6.

L.2 Benefits of FIR

L.2.1 State-of-the-art of Detection Mechanisms

As already underlined, the detection of faults is traditionally the method of choice to prevent fault attacks.

In the early years of fault tolerance in secure embedded systems, analogue solutions were used. They consist in disseminating voltage, temperature, light sensors or any miscellaneous combination thereof on the surface of the chip. The problem of this approach is that it requires a mixed design, which is much more complicated from a CAD perspective than a purely digital design. Also, the analogue parts are consuming a lot of power and area in the design. Those practical and economical reasons explain why the analogue solution is obsolescent.

Therefore modern designs resort to all-digital detection mechanisms. The generic ones exploit some artificial redundancy. It can be either implemented in time, space or information (code-based). All those strategies have been compared in [289], and shown to be roughly alike. Depending on the cryptographic scheme to protect, some dedicated countermeasures can also be implemented. The idea is to exploit some identities of the algorithm to protect so as to detect possible errors with a high probability. For example, in a typical encryption: the encrypted message can be decrypted and tested against the original plaintext. The same applies to digital signatures: the signature can be verified before being outputted. We wish to underline that these very verifications can represent

a weakness *per se*, notably in front of so-called *safe errors* attacks [477].

However, the resilience against faults attacks has seldom been proposed. At the opposite, resilience in observation attacks is definitely a hot topic. Following the proposal of Paul C. Kocher made at the rump-session of CHES 2006 [1] [245] to update the keys on a frequent and regular basis, ideas for side-channel attacks resilient schemes have come up, as illustrated for instance by the "Provable Security against Physical Attacks" workshop [269]. But, to our best knowledge, no investigation about resilience against fault injection attacks has been published so far. Actually, many techniques of reliability have been ported *as such* to security applications. Nonetheless the objectives of reliability and security do differ:

- Reliability requires ideally that either the computations are correct or that an alarm is raised;
- Security requires that the computation result, if erroneous, carries no information about secret involved in the computation. This is a more flexible requirement than for reliability. On the one hand, it allows the system to output a false result C^* instead of the correct one C, as long as it reveals no information about the secret K. A formalization of security models under fault attacks can be done, for instance taking example on the practice-oriented framework [434] in the sibling case of SCAs. Actually this work has already been initiated for instance by this preliminary paper [263]. From an information-theoretic perspective, the requirement can be stated as "*the mutual information between* (C, C^*) *and* K *is null*". On the other hand, rising an alarm can even be a vulnerability in some contexts. For instance, the differential behavior analysis (DBA [386]) manages to extract a key simply by knowing whether or not the computation went well, provided the fault model is of "stuck-at" type and roughly reproducible. FIR has no concept of alarm, hence is immune against such attack methods.

Therefore, in this paper, we challenge the reflex of transposing methods of reliability to security, because we prove that they are overly conservative.

L.2.2 Comparison between Detection and Resilience

Neither detection nor resilient schemes are able to withstand all the faults. Indeed, whatever the protection mechanism, we can theoretically build an attacker (possibly adaptative) able to replace an authentic value with another one. The goal of the countermeasure is to make this substitution very chancy.

In this subsection, we investigate the side-effects of the countermeasures. The detection strategy suffers two drawbacks [2] illustrated in Tab. L.1. First of all, the device can raise an alarm even if the result is correct. This is the case when the fault happens on a variable that does not impact the output. This situation is of course not true in general, otherwise the variable could have been removed from the implementation. However, in the course of a specific computation, this is indeed possible. One trivial example is the

1. Available on the CHES 2006 website.
2. In biometrics, the two drawbacks discussed in this paragraph would be called respectively *false reject* and *false accept* (whose rates are known as respectively "FRR" and "FAR").

Table L.1: Classical fault detection characteristics, where inconvenient features have been highlighted in red color.

		Ciphertext incorrect?	
		Yes	**No**
Alarm raised?	**Yes**	Safe	Problem of **availability**
	No	Problem of **security**	Safe

result of an `AND` gate, that has zero for one input, and that is faulted on its second input. The fault will not be propagated and the result will be correct irrespective of the fault taking place or not. However, if a detection mechanism raises an alarm, then the whole computation will be stopped and adequate actions will be undertaken, thus causing a denial of service (DoS) despite the absence of any security problem. The DoS can also be seen as an attack path, where the opponent's goal is simply to prevent the cryptosystem from functioning. Also, the detection process in itself can be threat. Unless implemented in a discrete manner, the detection process can be spied by the analysis of a side-channel [233, Chp. 2], thus opening the door to attacks exploiting hypothesis testing (*e.g.* safe errors [477]). Such attacks require nevertheless a lot of care, since the attacker must be able to cut the power of the system before it erases its secrets. The second drawback of detection mechanisms is that they do not cover all the possible faults, and some faults can propagate without being detected.

On the contrary, an ideal resilient scheme will feature:

- **an optimal availability**: false detections do not exist, since errors are not caught but propagated.
- **an optimal security**: the fault generates a wave of erroneous data independent of the previous pristine (and sensitive) values. Therefore no sensitive information is propagated.

Also, in terms of coding and deployment guidelines, the advantages of resilience as opposed to resistance (fault detection) are manifold. We can really claim that resilience is a new security approach to protect cryptography, because of these typical improvements:

- In traditional designs, miscellaneous checks are scattered in the code. For instance, ratification counters and baits are usual tricks to detect "blind attacks". No such extra operations are required in the context of fault resilience, since it is not catastrophic that the IC fails. To be perfectly clear, such subterfuges are more *palliative* than *curative*. They notably hinder automatic or formal code expertise, although some applications would demand such a high confidence evaluation level.
- When using detection, faults can also occur in the detection logic. But then, the

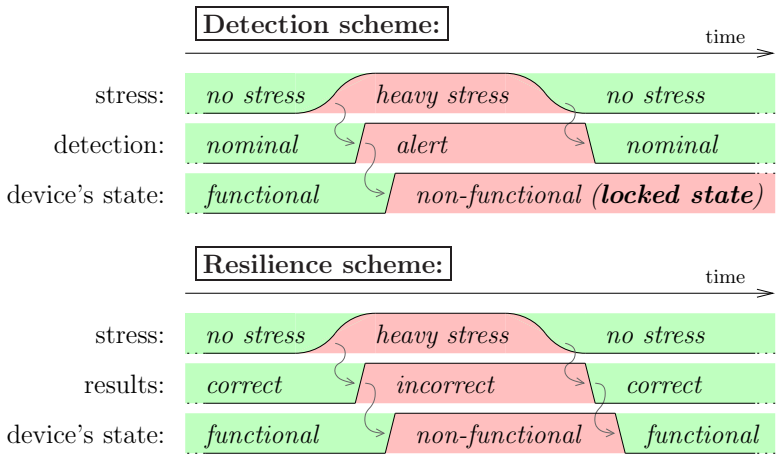


Figure 1: Suicide in case of fault detection (*top*), opposed to survival in case of fault resilience (*bottom*) protection schemes.

problem becomes eventually insolvable, since more and more logic is necessary (by recursion, we need detection logic for the detection logic, itself being protected by detection, *etc.*)

– On top of that, the resilience relieves the designer from having to deal with the reactions to the threat. These features are all in one very annoying for the chip manufacturer; if they are activated unexpectedly they possibly ruin the device, causing large costs to replace the defective card. Now, the secure chip manufacturers are often balancing between activating the maximum level of countermeasures and risking card auto-scuttling (false positive)[3]. Such a dilemma does not exist with fault resilience. The card starts to produce faulty results while under stress (either because of an attack or because of a natural hazard), but returns to its nominal operating conditions as soon as the stress disappears. Thus the risk of have a permanent damage due to a false alarm is merely nonexistent. This point is exemplified in Fig. 1.

L.2.3 Further Merits of the FIR

One feature that gives to FIR a remarkable strength is its agnosticism with respect to attacks. By making any faults independent at its source and during its propagation independent of the previous values, it merely prevents any attack at its root. Therefore, new scenario schemes not envisioned yet are thwarted proactively, which provides a for-

3. Remember that early countermeasures against faults were intended to make up for the poor quality card readers, that inappropriately injected unwanted electrical glitches in the smartcards! Also, Ross J. Anderson and Markus G. Kuhn explained in [11] that the wild fluctuations in clock frequency that frequently occur when a card is powered up and the supply circuit is stabilising, caused so many false alarms that the [detection] feature is no longer used by the card's operating system.

ward security. Typically, most – if not all – attacks studied so far are differential: they assume the attacker knows couples of correct & faulted computations corresponding to an identical (and presumably unknown) plaintext. Now, higher-order attacks could as well be possible: they would imply more than one faulty result. Additionally, faulted ciphertext-only attacks could also be devised. FIR fights all those future new threats that a pure DFA counter-measure would maybe fail to cover.

L.2.4 Related Works

Earlier publications have noticed the interest of allowing cryptographic devices to output faulty results, without jeopardizing their security. However, all those results focused on asymmetric cryptography, and more specifically on RSA. A fault tolerant RSA with CRT [4] algorithm is given and formally proved in [478]. This article introduces the concepts of "*fault infective CRT computation*" and "*fault infective CRT recombination*". the algorithm is designed to have the errors occurring during the "mod p" half propagate in the "mod q" half, and *vice-versa*, thus denying the Bellcore [49] attack. This idea is definitely a FIR, albeit crafted to the case of RSA and more specifically against the Bellcore attack, whereas in our paper, the FIR is algorithm-agnostic.

Other formal ways to secure sensitive algorithms have been proposed. For instance, the paper [137] about "Algorithmic Tamper-Proof" (ATP) explains how to protect an implementation, by the specification of security requirements on the circuit and by restricting the power of the attacker. A cryptographic module implementing the FIR is definitely not protected in the context described in paper [137]. We would like to make clear that the FIR notion introduced in our paper applies to a system that has a trusted environment: the asset at risk is therefore only the cryptographic core. In other terms, the two methods ATP and FIR do not consider the same security boundary.

L.3 Some Practical Implementations of FIR

The purpose of this section is to provide with some actual instances of resilient cryptographic schemes. For the sake of clarity, we focus on the protection of symmetric block encryption modules. Indeed, as they are deprived by construction from any algebraic properties, they are also the most difficult ones to protect. The state-of-the-art in asymmetrical algorithms protection is very well advanced and formally proved. An overview, on the example of RSA, can for instance be found in these papers [53, 54].

In the subsection L.3.1, we present a FIR approach that works at *high-level*, on top of an unprotected cryptographic module: it is a protocol-level resilient scheme. The subsection L.3.2 rather introduces two solutions at the *gate-level*, where FIR is intricated with the cryptographic module's implementation. In those two embodiments of FIR, we assume the cryptographic parameters are loaded securely, and thus that key alteration

4. The computations "mod $n = p \cdot q$" are done separately "mod p" and "mod q", and then combined back. This processing – possible only for the owner of the private key – speeds up the overall computation by a factor of four.

attacks (see for instance [137] or §III.C of [17]) are out of the scope. To sum up, our security goal is definitely the protection of symmetric cryptographic operations.

L.3.1 Formal Counter-Measures against Fault Injection Attacks

A differential fault analysis (DFA [45]) requires the same plaintext to be encrypted twice with the same key. Common attack scenarios consider the case where the attacker is able to inject one fault in only one of the encryptions. Then, she can deduce information about the key using a DFA. Thus, DFAs are made impossible if an attacker is not able to request twice the same encryption. It is possible to devise such a scheme, by making each new encryption unique thanks to nonce r, as typified by algorithm (1). Notice that the use of a random nonce in FIR implementations of symmetric encryption is similar to the use of a salt in user key derivation with password hashing technique.

Algorithm 1: Probabilistic Encryption Algorithm built on top of AES, non-protected against FIAs.

Input : A plaintext x to be encrypted with the key k, shared between the client and the server.

Output: A ciphertext along with a random number.

1 Determine a random number r of the same size as x; `/* This number will whiten` x `*/`.

2 Return the couple $(y = \text{AES}_k(x \oplus r), r)$.

This algorithm (1) is considered as secure against DFA because the probability that two encryptions are generated with the same plaintext is roughly speaking $2^{n/2}$, where n is the entropy of x or r. Indeed, this is a classical instance of the birthday paradox.

We mention additionally that the scheme of algorithm (1) protects against a broader class of attacks than only the DFAs. It is a random encryption scheme, that has the remarkable property that the attacker cannot decide if the encryption is actually faulty or not. Indeed, in an ideal block cipher, an attacker cannot distinguish between the outputs of that cipher and of a noise generator. Therefore, in the case of a random FIA, the attacker gets no additional information, hence no advantage, from her perturbations of algorithm (1). Thus, safe-error [477] attacks on the block cipher are also impossible: even if the attacker manages to inject a precise fault (in time, space and value) in the early rounds of the algorithm, there is no way for her to know from the encryption result whether this value is correct or not.

As a security notice, it must be understood that the protocol (1), used as such, can be forged. Indeed, if one authentic transaction is spied by an attacker, she gains access to a couple $(y = \text{AES}_k(x \oplus r), r)$. Now, let us consider the case where the attack wishes to impersonate the client. It is straightforward, in front of a new request m' to return a valid encryption without knowing the secret key k. The imposter can simply choose maliciously the random variable r' as $r' = m \oplus m' \oplus r$, and return (y, r'), which is a valid

encryption. Therefore, the protocol should include a challenge. For instance, the random variable r can be sent from the server instead of being chosen by the client itself.

Unfortunately, this scheme is not secure in decryption. As a matter of fact, the decryption algorithm corresponding to (1) is given in algorithm (2). This algorithm can be called repeatedly without the AES inputs being modified: it is deterministic.

Algorithm 2: Deterministic Decryption Algorithm matching algorithm (1).

Input : A ciphertext under the form $(y = \mathrm{AES}_k(x \oplus r), r)$ to be decrypted by the AES key k.
Output: The plaintext x.

1 Decrypt y with key k: $z = \mathrm{AES}_k^{-1}(y)$.
2 Return the demasked input: $z \oplus r = x$.

This situation can however be exploited to protect low cost embedded systems, such as smartcards or RFID tags, that communicate with a larger device, such as a reader. In this situation, there is a natural asymmetry between the two protagonists. This fact has been emphasized in other publications on lightweight embedded systems security, such as [152, §1]. It is fairly easy to protect the reader against fault attacks by "physical tamper-proof measures". For instance, the reader electronic circuits can be imprisoned into a mold, protected with a pasted metallic cover and sealed into a box equipped with intrusion detection sensors. The same level of sophistication is impossible for smartcard or tags modules, because their form factor is extremely constrained in size (due to stringent requirements about the mechanical strength edicted by standard ISO 7816-1). Hence ways to attack smartcards are – unfortunately – very numerous [251]. Additionally, smartcards are cheaper to buy than readers, and, to top it all, the selling of smartcards is necessarily less restricted than that of readers, because in any deployment context, there are more smartcards out than card readers. Therefore, the attacker will most certainly prefer to attack the embedded system to extract the shared secret key. Thus, if the reader plays the decryption (2) and the embedded system the encryption (1), the unbalance between the tamper-resistance of the two devices is made up by the opposite unbalance of the algorithm, in terms of resistance against DFA. This strategy of reinforcing the security by algorithmic means of the weakest element in the security chain is illustrated in Fig. 2.

Notice that if a handy homomorphous encryption algorithm HEA is available, a completely secure encryption/decryption scheme can be devised. Let us denote by $\mathrm{HDA} = \mathrm{HEA}^{-1}$ the corresponding decryption algorithm and $\times$ the composition law in the group of homomorphy:

$$\forall y_1, y_2, \quad \mathrm{HDA}(y_1 \times y_2) = \mathrm{HDA}(y_1) \times \mathrm{HDA}(y_2)\,.$$

The encryption proceeds as per algorithm (1) using HEA instead of AES, whereas the decryption consists in algorithm (3). This scheme can use for instance Paillier's cryptosystem [350] as underlying encryption primitive. However care must be taken with

Figure 2: Probabilistic encryption is performed on the most vulnerable device while the deterministic decryption is safely carried out within the most secure device.

RSA [49].

Algorithm 3: Probabilistic Decryption Algorithm matching (1) with HEA instead of AES as underlying cipher.

Input : A ciphertext under the form $(y = \text{HEA}_k(x \oplus r), r)$ to be decrypted by the HEA key k.
Output: The plaintext x.

1 Determine a random number s of the same size as y or r.
2 Return $\text{HDA}_k(y \times s)/\text{HDA}_k(s) \oplus r = x$.

The resilient algorithms presented in this subsection L.3.1 have the drawback that the size of the ciphertext is doubled. This can be a limitation for instance in contactless cards authentication, where the transmission time must remain short. Also in wireless sensor network the increase of the data transmitted means a very high cost in term of power.

Nonetheless the algorithm (1) can be made more bandwidth and power-efficient if the message x to encrypt is cut in several blocks. In this case, alternative encodings, such as the probabilistic all-or-nothing transform (AONT) described in [303, 302], could be taken advantage of. This paper and this patent introduce a probabilistic symmetric encryption algorithm, in a view to thwart SCAs. With respect to other probabilistic symmetric encryption scheme (most of the times, the encryption involves a random IV – which is short for *initialization vector*), this AONT scheme is original in the sense that the randomness is not disclosed along with the ciphertext. This denies the possibility to conduct a side-channel attack on the first round(s) of the encryption algorithm. A similar scheme has also been described in [304]. As such, this all-or-nothing scheme (in general, but also under the form of its "Probabilistic Signature Scheme", *aka* PSS, avatar [90]) is an implementation of FIR. In addition, it reduces the number of blocks to be exchanged to the number of plaintext blocks plus one. In summary, algorithm (1) combined with [303] has the benefit of bringing a SCA-resistance in addition to the FIA-resilience. Certainly, this suggestion of protocol-level countermeasure can be optimized, but we leave this topic open for future works [31].

L.3.2 Multi-Valued and Redundant Representation Logics

Multi-valued logics allow to encode more than one bit with one electrical state. It is for instance used in some power-constant logic styles [14]. Let us consider the case of an equipotential holding three states, denoted `0`, `1/2` and `1`, amongst which only the two `0` and `1` are functional. Then, if a fault turns a valid value into `1/2`, the provenance state (either `0` or `1`) has been forgotten.

The same goes for redundant logics, such as the m-out-of-n representations (for $0 < m < n$). For instance, the 1-out-of-2 representation, also known as dual-rail with precharge logic (DPL), admits two valid states, denoted by `01` and `10`, and two invalid

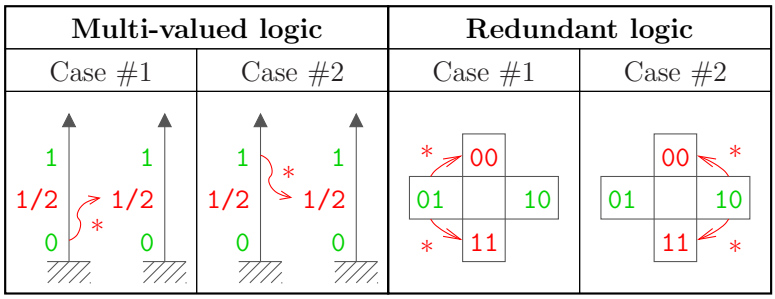


Figure 3: Two kinds of faults (in red), namely $\{0,1\} \xrightarrow{*} 1/2$ for 3-valued logic and $\{01,10\} \xrightarrow{*} \{00,11\}$ for DPL, after which the initial value (in green) has been forgotten.

states, denoted by 00 and 11. In the case one fault turns a valid token into an invalid one, the value before the fault is lost. The effect of faults on these two logic styles is summed up in Fig. 3. It clearly appears that the state after the fault is decorrelated from the initial state, thereby establishing the resilience, for the relevant cases where the data is sensitive.

This protection mechanism is nonetheless less powerful than that based on input message randomization. Indeed, by merely looking for invalid tokens in the output, the attacker can decide if a fault has been having an effect. Without loss of generality, let us take the example of DPL where the spacer token used for precharge is 00. One typical fault scenario is the valid tokens not having enough time to evaluate. Hence, some tokens supposed in theory to get regular value 01 or 10 remain in practice stuck at idle value 00. If the ciphertext has an abnormally low Hamming weight, the attacker can deduce that, with a high probability, a fault has been injected successfully and has propagated. Thus redundant logic styles are not secure against all kinds of attacks that do not exploit the faulted result, but merely the behavior (faulty or not). This concerns the safe-errors attacks [477], the DBA [386] and the FSA [266]. Only DFAs are thwarted, because the value of the faulted ciphertext is unrelated to the netlist internal secrets.

Now, the resilience only works in the case the attacker fails to inject "valid false" faults, *i.e* $0 \xleftrightarrow{*} 1$ faults in multi-valued logic or $01 \xleftrightarrow{*} 10$ faults in DPL. Let us assume, for the moment [5], that this situation is rare. It seems all the more difficult to achieve in DPL because the attacker must produce two antinomic concerted faults.

As will be exposed into greatest details in Sec. L.4, the resilience will build up each time a valid false is produced along with invalid faults. In this case, the two faults will propagate, and if the logic favors the generation of invalid instead of valid states, then the diffusion of the netlist will encourage the invalid states to hide the false valid states. This case is optimal if the logic meets this requirement:

> "if any input is invalid, so is the output".

This behavior is "saturating"; the faults will percolate in the netlist and the invalid values

5. A study of "valid false" survival conditions is provided in Sec. L.4.5.

will saturate most of the nets, thereby absorbing all the false valids that are crossed. So the resilience is amplified by the diffusion in the netlist and the collaborative behavior of gates to favor invalid values propagation. This phenomenon of invalid values (dominant) suppressing false valid values (recessive) is further detailed in the next section L.4.

L.4 Dual-Rail with Precharge Logic as a Global Counter-measure against Implementation-Level Attacks

DPL styles are solutions primarily designed to protect a cryptographic implementation against side-channel attacks. However, it has been noticed that these styles can also natively withstand some perturbation attacks [318, 319, 411, 33]. It has already been underlined in Sec. L.2 that, unlike traditional counter-measures against fault attacks, the DPL does not implement a protection, but is rather resilient. This means that faults are not caught, but rather left free to cascade their effect, knowing that eventually their observable consequences will not be harmful from a security standpoint.

L.4.1 Requirements for Simultaneous SCA and FIA Protection

In order to better illustrate the close relationship between observation and perturbation attacks, we need to notice that security perimeters depend on the application. For instance, in an ISO/IEC 7816 compliant smartcard, several security violation situations can be encountered.

- The critical part is the memory in case of an external authentication. Indeed, if the memory can be corrupted, then any rogue reader can be forced to be seen as authentic. Here, there is no secret to retrieve, but simply an invalid state to be setup by force.
- However, during an internal authentication, the smartcard uses its cryptographic secret. Therefore, the risk for the smartcard is to have its key retrieved illegitimately. Differential fault attacks and side-channel attacks are two tools available to recover the key. In addition, as the protection against attacks is costly, the designer will try to partition the cryptographic block at risk. Typically, when he implements symmetrical encryption, this block can be split into:
  - a control part, subject to fault attacks, such as round reduction attacks [316], but leaking no sensitive information as the algorithm is supposed to be known by the attacker (common assumption with Kerckhoffs' law), and
  - a data processing part, subject to both fault attacks, such as DFAs [45, 357], and side-channel attacks, such as DPA [248].

The overall requirement for security against implementation-level attacks in a smartcard is depicted in Fig. 4. This block-diagram shows in red the security boundary for fault attacks and in cyan that for SCAs. It appears clearly that some organs shall be protected only against fault attacks, but that all the organs that shall be protected against SCA must also be protected against FIA. This is an advanced question, all the more important as it is in this part of the design that the largest overheads are expected.

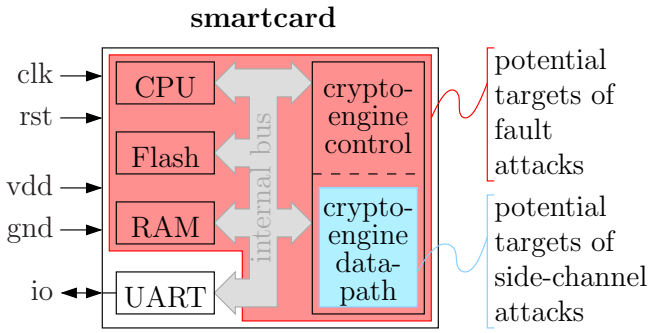


Figure 4: Susceptible organs of a smartcard in two representative sensitive operations (`EXTERNAL AUTHENTICATE` and `INTERNAL AUTHENTICATE`). Typically, the cryptography will be triple-DES or AES, *i.e.* the most widely industrially adopted block ciphers.

The countermeasures against SCA include:
- information hiding, implemented with DPL,
- information masking, implemented with random splitting of data into shares.

More information about these two categories of protection against SCAs can be found in the "DPA book" [290], respectively at chapter 7 and 9. Amongst this array of possible protections, DPLs [419, 97] are of particular interest because they have native protections against DFAs. We will thus focus in the rest of this article on the combined DFA and SCA protection of the datapath of cryptographic modules; The type of fault attacks we consider are those described in [144], the two most famous of them being that of Biham & Shamir [45] (DES [336]) or Piret & Quisquater [357] (AES [337]), enhanced by Tunstall in [463]. Another motivation to focus on the crypto-datapath is that it is usually the most complex design part; therefore it represents the largest area of the design and contains the longest critical timing paths. This explains that local faults are more likely to target the datapath because of its predominant surface, and that global faults also affect preferentially the datapath that is most tight when it comes to meeting the setup time constraint.

L.4.2 Previous Art about DPL in the Presence of Faults

We use the following notations for the DPL representation. Every logical variable a is represented by a couple (a_f, a_t) of wires, that carry two values. The term a_f (resp. a_t) is short for the proposition "a is false, *i.e.* $a = 0$" (resp. "a is true, *i.e.* $a = 1$"). The semantic of the four possible combinations is detailed below.
- a is VALID if $\boxed{a_f \oplus a_t = 1}$. More precisely, VALID $\doteq \{\text{VALID0}, \text{VALID1}\}$ or, more explicitly,
  VALID $\doteq \{(1,0),(0,1)\}$.
- a is NULL if $\boxed{a_f \oplus a_t = 0}$. More precisely, NULL $\doteq \{\text{NULL0}, \text{NULL1}\}$ or, more explicitly,

NULL $\doteq \{(0,0),(1,1)\}$.

The two NULL states are used alternatively with the VALID ones as precharge stage, so that the next evaluation starts afresh from a known state. The DPL protocol is recalled in Fig. 1.14.

There are two flavors of DPL, depending on whether they feature the early propagation effect (named EPE in the literature, and incidentally discovered independently by [439] and [253] in the same year) or are protected against it. The definition of those variants can be summarized by the following conditions to be fulfilled by all the instances f:

- **DPL w/ EPE**: $\exists a$ VALID, $f(a, \text{NULL}) = \text{VALID}$;
- **DPL w/o EPE**: $\forall a$ VALID, $f(a, \text{NULL}) = \text{NULL}$.

To be properly protected against SCAs, those logics must be balanced at the layout-level [457, 191, 174], to preserve indiscernibility properties (typically true $\leftrightarrow$ false symmetry). Otherwise, straightforward attacks that exploit the physical unbalance become possible [395]. However, when analyzing those logics regarding FIR, the physical layout can be forgotten, and only the logical functions are considered.

In DPL, only results on *evaluation* are observable, because *return to precharge* faults are not outputted. We adopt the following faults typology on DPL:

- **Asymmetric faults**: {VALID0, VALID1} $\xrightarrow{\downarrow}$ NULL0, triggered by **global** perturbations (*e.g.* caused by a setup time violation due to power/clock glitch, overclocking or under-powering);
- **Symmetric faults**: {VALID0, VALID1} $\xrightarrow{\downarrow \text{ or } \uparrow}$ {NULL0, NULL1}, triggered by **local** perturbations (*e.g.* caused by injection of high energy laser light, electromagnetic field or particles beam).

L.4.2.1 *DPL w/ EPE* is Protected against Multiple *Asymmetrical* Faults

WDDL [456] is a typical DPL w/ EPE style. In this logic, the `AND` function is defined as: $(y_f, y_t) \doteq (a_f + b_f, a_t \cdot b_t)$. We use the following color code in Boolean truth tables:

- gray: the regular truth table in the absence of faults (*i.e.* the intended functionality),
- purple: anticipated values (evaluation even if not all inputs are valid).

Otherwise, the green and red colors still represent respectively correct and incorrect behaviors or properties.

As shown below, WDDL can propagate correct valid results in the presence of asymmetrical faults.

b \ a	VALID0	VALID1	NULL0
VALID0	VALID0	VALID0	VALID0 (**EPE**)
VALID1	VALID0	VALID1	NULL0
NULL0	VALID0 (**EPE**)	NULL0	NULL0

This behavior is positively resilient. It is that of the `U`nitialized value in VHDL enumerated type `ieee.std_logic_1164.std_ulogic`, recalled below:

b \ a	'O'	'1'	'U'
'O'	'O'	'O'	'O'
'1'	'O'	'1'	'U'
'U'	'O'	'U'	'U'

where the tokens {VALID0, VALID1, NULL0} implement respectively the items {'O', '1', 'U'}.

These conclusions can be challenged in the case of a coupling of the fault injection analysis with a side-channel analysis. For instance, the fault sensibility analysis (FSA [266]) can, under some circumstances, exploit the unbalance within the two wires making up a dual-rail pair. However, the FSA has only been demonstrated as partially successful on a WDDL chip; and WDDL is known to be extremely unbalanced [395].

Actually, this FIA-resistance solution has already been sketched in [230]. This article introduces two methods to protect circuits against FIAs.

The first one consists in resisting to an arbitrary number of "stuck-at-0"[6]. Those "*reset faults*" correspond to our "*asymmetric faults*". However, this publication is overly conservative; invalid tokens are generated even if the data is not tainted. Also, the authors of [230] add a series of cascade gates at the output of the circuit. Their role is to turn all other valid tokens to invalid ones. Additionally, they request that the circuit commits suicide at this point (when the ciphertext is all NULL, noted "$\perp$" in [230]). Our key remark is that those two requirements are actually overkill. Indeed, the overall security is not jeopardized if some valid and some invalid tokens are outputted; therefore, we can save the cascade stage. In addition, we insist that it is then useless to permanently destroy the circuit: as we know the attacker only gets faulted crypto results that do not convey any information about the sensitive variables, it is safe to continue without erasing the secrets, that are merely not compromized. Therefore, the scheme we present is more user-friendly, in the sense it keeps the application up-and-running unless a fault is indeed influencing the result.

The second countermeasure against arbitrary faults in [230] is more *ad hoc*, since one needs to know the maximum number of faults an attacker can inject to dimension the level of protection (based on an adaptively sized countermeasure). In the next paragraph, we study FIR in the presence of multiple symmetric faults.

L.4.2.2 *DPL w/ EPE* is not Protected against Multiple *Symmetric* Faults

To start with, we assume neither $a \xrightarrow{*} \overline{a}$ nor $b \xrightarrow{*} \overline{b}$ happens. However, even in this favorable case, WDDL can generate incorrect false results. They are presented by skulls (symbol: ☠) in the following table.

b \ a	VALID0	VALID1	NULL0	NULL1
VALID0	VALID0	VALID0	VALID0 (**EPE**)	VALID0 (**EPE**)
VALID1	VALID0	VALID1	NULL0	NULL1
NULL0	VALID0 (**EPE**)	NULL0	NULL0	VALID0 (☠)
NULL1	VALID0 (**EPE**)	NULL1	VALID0 (☠)	NULL1

6. ...or equivalently "stuck-at-1" for all the faults.

For instance, the twain simultaneous errors:

1. $a = \text{VALID1} \xrightarrow{*\uparrow} a = \text{NULL1}$ and
2. $b = \text{VALID1} \xrightarrow{*\downarrow} b = \text{NULL0}$

trigger a dreadful transformation: $\text{VALID1} \xrightarrow{*} \text{VALID0}$.

Therefore, because of EPE, logical inversions $f(a,b) \xrightarrow{*} \overline{f(a,b)}$ can occur, which makes FIAs (such as DFAs) possible.

L.4.2.3 *DPL w/o EPE* is Protected in front of Multiple *Symmetric* Faults

Now, the DPL w/o EPE styles are protected against multiple symmetric (hence asymmetric) faults. This is shown in the table below.

b \ a	VALID0	VALID1	NULL0	NULL1
VALID0	VALID0	VALID0	NULL0	NULL1
VALID1	VALID0	VALID1	NULL0	NULL1
NULL0	NULL0	NULL0	NULL0	NULL1
NULL1	NULL1	NULL1	NULL0	NULL1

Remark that if we call:

- `'0'`: VALID0,
- `'1'`: VALID1,
- `'X'`: NULL $= \{\text{NULL0}, \text{NULL1}\}$,

then we have the same behavior (*i.e.* "*propagate always*") as VHDL. This is illustrated below:

b \ a	`'0'`	`'1'`	`'X'`
`'0'`	`'0'`	`'0'`	`'X'`
`'1'`	`'0'`	`'1'`	`'X'`
`'X'`	`'X'`	`'X'`	`'X'`

Finally, we note that even if a few mutations $a \xrightarrow{*} \overline{a}$ exist for some variables a, it is very likely that the `'X'` wave caused by $a \xrightarrow{*} \text{NULL}$ *eats* them. As detailed in the next sub-section, the recessivity of `'X'` over NULL, coupled with the avalanche of `'X'` caused by the diffusion property of the logic, accounts for that.

L.4.3 Revisiting the Comparison Resilience vs. Detection

One can argue that the DPL used as a FIR is in fact a very low-grain fault detection scheme. Indeed, FIR shares with the detection strategy the fact that redundancy is required. However, it is coupled to a diffusion that makes the detection at one stage take advantage of the rest of the stages. This detection is propagated in a wave, that constitutes a collaborative strategy that is absent from the pure detection schemes. This difference is illustrated in Fig. 5. In traditional detection schemes, the computation (noted: C) and the detection (noted: D) logics are dissociated. In particular, the detection blocks do not communicate. In the DPL FIR scheme, the computation and the

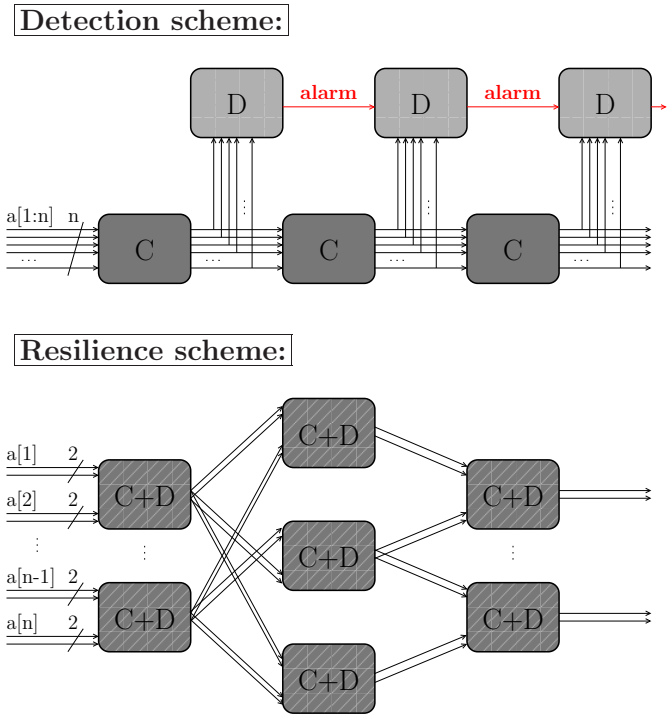


Figure 5: Difference of detection and resilience working factors, represented on an example netlist.

detection are merged (noted: C+D) and this information propagates downwards the netlist.

There are two properties of DPL that help resilience:

- The **redundancy** of the netlist. At an n-bit output of a combinational block, only 2^n amongst the 2^{2n} possible ones are valid.
- The **diffusion** within the netlist, which is characteristic to the cryptographic algorithms. This property is especially true at the netlist level for logics free from EPE [33]. Indeed, the fanout of each gate is double w.r.t. separable logics such as WDDL [456].

Current detections schemes work independently of the computation and in a non-collaborative way. At the opposite, FIR consists in intricating the detection agents with the computation and to tightly interconnect them. The objective is to trigger a proliferation of tamper-evidence logic markers (NULL tokens).

L.4.4 Cost Estimation of FIR versus Traditional Approaches

The traditional approach to counteract implementation-level attacks is a composition. The recommendations formulate like this:

– first use detection schemes, that can be inserted early at the RTL of the algorithm [260];
– then map this FIA-aware RTL description into a SCA-proof logic style. Indeed, the detection logic manipulates sensitive variables, and might itself leak secrets [380]. Therefore, it deserves a protection against SCAs. In a similar fashion, the study reported in paper [287] confirms that gate-level countermeasures against FIAs do not reduce the information leakage.

This implies that the overhead of the FIA and SCA countermeasures get multiplied.

A typical overhead for FIA countermeasures can be found in [289]. Let us consider the case of a non-linear code, such as [237], that is suited to detect multiple faults. Its overhead is 77 % in area and 15 % in throughput.

As such, those performance losses are more affordable than those required to thwart SCAs. For instance, WDDL incures an increase of 3.1 in area and 3.9 in throughput [453].

The combination of [237] and [453] results in an increase of 5.5 in area and 4.5 in throughput.

Those results are to be contrasted with the FIR approach using an EPE-proof DPL style. This style already merges FIA and SCA countermeasures. The reported overheads for two of those logics are given in Tab. L.2. It clearly appears that using a symbiotic SCA+FIA countermeasure is more efficient than combining two countermeasures one on top of each other.

We notice that those alternative "DPL without EPE" logics yield similar performances: DRSL [79] [7], iMDPL [358], IWDDL [301], STTL [417, 418], SecLib [193, 175, 189, 210], WDDL w/o EPE [33, 37], BCDL [331, 93] and LBDL [481].

We also attract the reader's attention on the fact that asynchronous logics, especially the quasi-delay insensitive (QDI) style [318, 316], can be implemented in DPL [171]. Now, asynchronous logic is designed to remain functional irrespective of the environmental variations. Concrete work [480] on this topic had been carried out in the framework of the G3Card project [132]. However, the G3Card consortium only detects NULL1 as an error marker in a DPL protocol where the only allowed spacer is NULL0. This signalization is restrictive and do not consider propagation of errors; instead, an instantaneous detection is suggested, which seams hard to put in practice in *real-time* given that such checks shall be done for each and every gate of the design. Moreover, asynchronous QDI logics have a drawback in terms of resilience: each gate being sequential in nature (due to the necessary handshakes with the upstream fanin and downstream fanout gates), a fault can cause a deadlock, should the fault cause a protocol violation (*i.e.* the transitions depicted in Fig. 1.14 are not respected). To relieve the circuit from this deadlock, the asynchronous circuit shall be reset. Thus the resilience provided by an asynchronous circuit is in-between the two cases illustrated in Fig. 1. The card is not destroyed permanently, since a reinitialization relaunches it; however, the system must detect that the logic hung (perhaps with the help of a watchdog) in order to restart it. Despite of these discrepancies with the FIR concepts, we note that QDI still increases the number of situations where the circuit remains functional, while remaining "resilient" if the external conditions are

7. DRSL is however shown to have a built-in security flaw in [331].

Table L.2: Performance overhead of different SCA+FIA countermeasures.

Strategy	**Detection + DPL**	**Resilience = DPL**	
Countermeasure	[237] + [453]	DRSL [79]	IWDDL [301]
Area	5.49 ×	2.56 ×	4.34 ×
Throughput	4.49 ×	2.00 ×	1.53 ×

too harsh.

Eventually, we wish to underline that these overheads are not that dramatic when contrasted with those encountered in other domains that also require dependability features. Typically, the avionic industry makes use of techniques such as triple modular redundancy (TMR) to thwart single event upsets (SEUs). An example of a memorization element in TMR style is given in Fig. 6. The amount of logic involved in this structure is by far larger than that required in the DPL counter-part, depicted in Fig. 7. This structure has two stages to accompany the evaluation $\leftrightarrow$ dynamic of the DPL protocol. We notably insist that such a construction is naturally immune to the attack presented in [320], that exploits an optimization of some DPL style: when the redundant dual-rail state is stored as one single bit, an exploitable leakage appears at the flip-flop level. To conclude this comparison between figures 6 and 7, we emphasize that the overhead figures shall not be considered in absolute, but relatively to the protection goal that is intended to be achieved.

L.4.5 Associating Three Protections to Reduce the Probability of a Successful FIA

Some faults in DPL circuits do not disclose any information about the faulted sensitive variable. However, in the case false valid are generated, the problem becomes different. This can happen in two problematic cases:

1. When the absorbing fault is too deep in the logic cone w.r.t. the false valid, as shown in Fig. 8, where f is a block with perfect[8] diffusion, such as a substitution box implemented in logic. In this case, if the logic cone covered by the `'X'` happens to yield a correct value, then a valid fault is generated; unless the `'X'` are checked for at the output.
2. When a valid false occurs on one column alone, but that an `'X'` is generated on another column (knowing the two columns are not interfering in AES last round). In this case also, the faulty behavior can be observed by checking the validity of all the output bits.

To fight these remaining risks, three protections can be associated so as to increase the security level:

8. Understand: as "close to perfect" as Boolean functions of finite dimensions can offer.

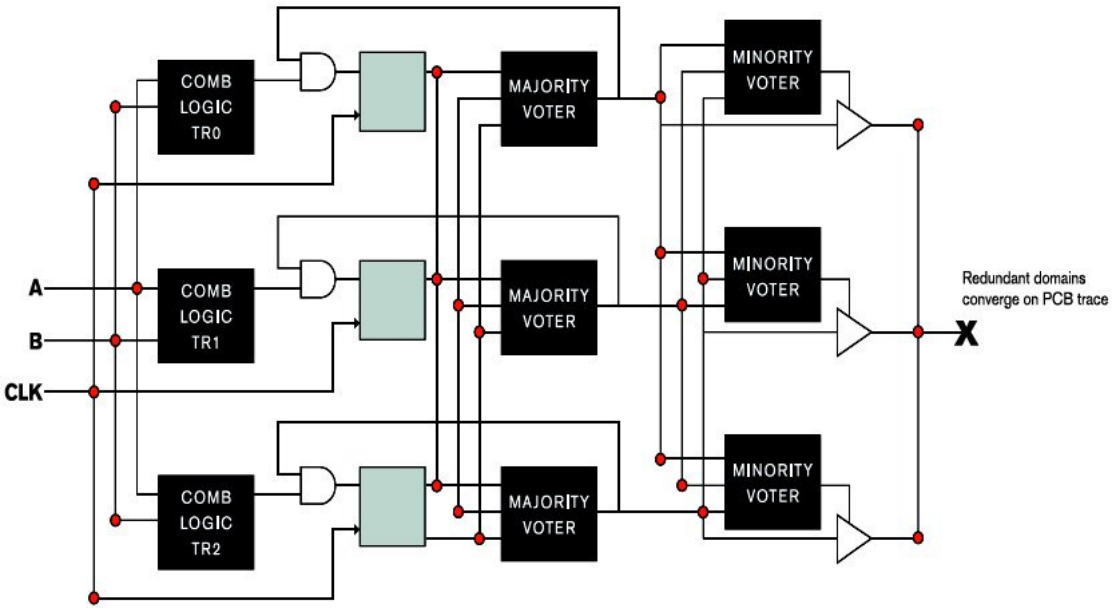


Figure 6: Memorization element in triple modular redundancy as implemented in Xilinx "XTMR" solution [448].

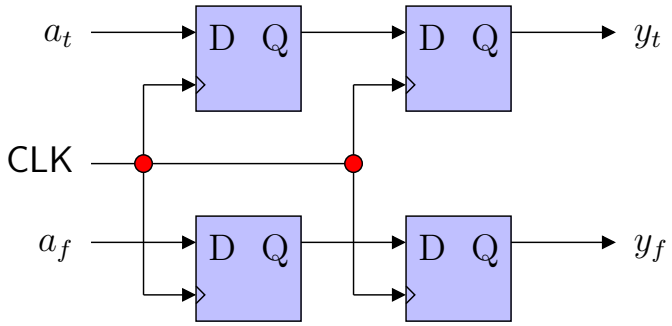


Figure 7: Memorization element in DPL; although four times larger than an unprotected flip-flop (DFF, represented as a violet square), this structure is nevertheless much smaller than that involved in TMR logic (see Fig. 6).

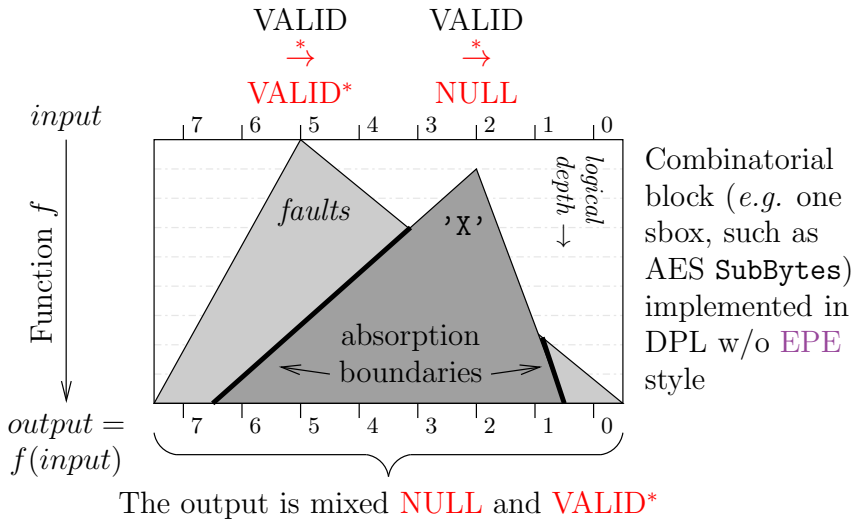


Figure 8: Multiple faults, where the false valid is not completely hidden by the `'X'` wave. The `'X'` avalanche absorbs most, if not all, the valid faults.

1. DPL, as detailed in the previous section.
2. Test for the existence of NULLs at the end of each computation. This sanity check basically consists in evaluating the Boolean security flag $\prod_{y \in \{\text{outputs}\}} (y_t \oplus y_f)$ [98].
3. Regular detection schemes, such as coding.

L.5 Applicability of Resilience with Certification Procedures

The two main certification schemes of security products are the FIPS 140 and the common criteria. We examine in this section if the resilience can be applied with the current version of those standard, or if the standards are too conservative.

L.5.1 NIST FIPS 140-3

The FIPS 140 [367, 368] formulates security requirements for cryptographic modules. It defines four levels of security, the highest of which is referred to as "security level 4". The functional security objectives of FIPS 140 are defined in §3. It includes those two requirements:

1. to detect errors in the operation of the cryptographic module and
2. to prevent the compromise or the modification of sensitive data and SSPs (Sensitive Security Parameters) resulting from these errors.

The "resilience" protection discussed in this article definitely fulfills the second requirement. However, not all resilient schemes comply with the first requirement. For instance,

using the randomized homomorphic encryption (Algorithm 1), the errors cannot be detected. The partial resilience of dual-rail type countermeasure can allow a detection of the fault. However, the security of this scheme is ensured even if there is no detection. This means that FIPS-140 standards 2 & 3 are not resilience-ready, although they express this idea.

More precisely, the exact statement of the requirements is detailed in §4.5.5 (140-2 [367]) or §4.6.5 (140-3 [368]). For the security level 4, the cryptographic module shall either employ environmental failure protection (EFP) features or undergo environmental failure testing (EFT). The EFP consists in a constant monitoring of the environment (temperature and voltage) whereas EFP is an a priori characterization of the perturbation consequences. In both cases, the protection circuitry shall either (1) shutdown the module to prevent further operation or (2) immediately zeroize all plaintext secret and private cryptographic keys and SSPs.

Such authoritative and irremediable actions could have been prevented using a resilience scheme, without compromizing the device security. Therefore, we find that FIPS 140-{2,3} standards are too strict, resulting in potential inconveniences from the user perspective if non malicious faults cause the module shutdown or zeroization.

L.5.2 Common Criteria

The Common Criteria (CC) [1] is a framework that permits comparability between results of independent security evaluations. It is an international standard ISO/IEC 15408:2005. The CC in themselves do not specify security requirements. Instead, a "target of evaluation" (TOE) must meet "security targets" (ST). Zero, one or more "protection profiles" (PP) must be respected by the ST. However, for marketing reasons, in practice, the ST complies to at least one PP. The security requirements are expressed in the PPs, whose structure is standardized but whose content is up to the designer. This flexibility allows a designer to tailor the PP to his (or that of his client) security objectives. Therefore, the CC readily accepts the resilience as a solution against fault attacks.

L.6 Conclusions and Perspectives

In embedded devices, fault attacks are usually combated in software. The dominant strategy is their detection, which is costly and non-exhaustive. We present in this paper an approach based on resilience. The faults are not necessarily captured, but the information they contain about any secret is nullified. The benefits of this approach are the ergonomy and the cost. First of all, the resilience impose no destruction of the secrets in case of a fault attack; thus, in case of natural (non-malevolent faults) the user experience is a transient DoS, as opposed to a permanent DoS in traditional detection-based countermeasures. Symmetrically, when a fault is injected successfully but has no consequence in the computation, a card protected with a detection-based scheme may react, whereas this inconvenience is nonexistent in the resilience-based scheme. Several concrete

methods to implement resilient symmetrical encryption are proposed, amongst which a random mode of operation that is suitable for low-cost (without expensive module-level protections) smartcards. When the designer can propose a hardware counter-measure, we suggest the use of multi-valued or DPL styles. Those logics simultaneously protect against observation and perturbation attacks, and are cheaper than detection based on codes.

As a perspective, we intend to quantify the optimal parameters of code-based detection schemes that can be added to a DPL logic (evoked in Sec. L.4.5) to further reduce the number of faulty results outputted by the device. Also, we thrive to define a formal framework based on the information theory that could describe with commensurable metrics the resistance of a cryptographic implementations to both SCA and FIA.

Acknowledgments

The authors are very grateful to the five anonymous reviewers, that all contributed to improve the paper and to better place it in its scientific context. Novel ideas have also be suggested, that all open the door to efficient and formally proved countermeasures against active and passive attacks. We also thank the positive inputs received from the audience during the presentation at FDTC 2010 in Santa Barbara, especially from Jean-Christophe Courrège, Guido Bertoni and Matthieu Rivain.

Appendix M

Performance Evaluation of Protocols Resilient to Physical Attacks

Extended version of article [209]

Authors: *Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Nidhal Selmane and Denis Réal*

Abstract

Cryptographic implementations are vulnerable to physical attacks. Many countermeasures to resist them have been proposed in the past. However, they are all specific to a given attacker and allow to mitigate the risk only up to a certain level: improved attacks on those countermeasures can most of the time be devised. Therefore, a new trend consists in making cryptographic implementations resilient to physical attacks. This strategy makes it possible to prove the countermeasure against all possible types of attackers captured by a security model. Several resilient schemes for the protection of block ciphers exist. For a given security objective, they all permit to reach the same security level. Therefore, they differentiate only according to their efficiency. We first show that the genuine versions of these protocols achieve different I/O bandwidth and computational performance. Our second contribution is to improve those protocols thanks to a message blinding, assuming passive attacks require more than two traces to be successful. Then, we bring as a third contribution the fact that the improved versions of the protocols are very much alike, and that the difference between them depends only from the specific details of their instantiation.

Keywords: Implementation-level attacks; Symmetric Encryption; Resilience (against Passive & Active Attacks); Performance.

M.1 Introduction

Implementation-level attacks aim at retrieving secrets concealed in embedded devices. They consist in passive attacks, where the attacker records some physical side-channel leaked out of the circuit, and active attacks, where the attacker perturbs the circuit in a view to have it output incorrect results. Unless the circuit features dedicated countermeasures, an attacker manages to exploit the leakage or the errors to successfully retrieve the key.

For a long time, engineers have attempted to make such a key retrieval harder, by essentially removing the dependence between the inner secret and the leakage by making the side-channel constant or random, and by detecting any fault injection. However, it has appeared that if those techniques manage to increase the attacks difficulty, they do not totally prevent them. Second order flaws have appeared and combined attacks also have shown up. Thus, assessing the exact security gain conveyed by those techniques has become hard.

For this reason, a recent trend has been to promote provable countermeasures, for which a rationale of attack impossibility can be demonstrated. The *resilience* strategy to thwart passive and active attacks meets this requirement. The advance of this technique is to allow a circuit to leak information and to output incorrect results, as long as they do not compromise the keys. In asymmetric cryptography, such schemes are already known [90]. Now, protecting symmetric cryptography remains a challenge, because block ciphers are less structured.

The rest of the article is structured as follows. The known resilient schemes are described in Sec. M.2. As those schemes were initially presented in different adversarial models, we mention in Sec. M.3 the plurality of the existing risks and settle a common security objective. Within this shared security framework, the various protocols resilient to physical attacks differ only by their performances. To compare their relative performances, we consider in Sec. M.4 two scenarios that correspond to two typical use cases. Then, in Sec. M.5, we introduce a novel resilient protocol that takes the most of the session keys while remaining secure against both passive and active implementation-level attacks. It trades some cryptographic-grade primitives for lower cost primitives, thereby increasing the performances beyond the state-of-the-art without jeopardizing the security. Eventually, Sec. M.6 concludes the paper and opens some perspectives, notably on the need for implementation-level robust lightweight primitives, that can be instantiated advantageously in low cost but highly secure embedded devices, while maintaining the formality of the security analysis.

M.2 State-of-the-Art

Several resilient computation schemes have emerged recently. Most of them are "leakage-resilient", *i.e.* resilient against passive side-channel attacks, that consist merely in observing the physical emanations leaking from a cryptographic device. Some *ad hoc* constructs (*i.e.* not based on standardized algorithms, such as AES) for leakage-resilient

stream ciphers [355] and signatures [124] have for instance been described.

However, many industrial applications require the cryptosystem to have its security based on standardized primitives. We therefore concentrate in the rest of this article on the protection of a block cipher g_k, such as the advanced encryption standard (AES), against physical attacks aiming at recovering its secret key k. To our best knowledge, only four publications tackle with protocol-level resilience for block ciphers. Three of them, namely *indexed key update* (abridged IKU and detailed in Sec. M.2.1), *fresh re-keying* (abridged FRK and detailed in Sec. M.2.2) and *all-or-nothing encryption* (abridged AONE and detailed in Sec. M.2.4), deal with leakage resilience. The second protocol, FRK, is described for single block encryptions. As will be discussed in Sec. M.3, under this limitation, IKU and FRK can be proved resilient to active attacks. At the opposite, the fourth protocol, called fault injection resilience (abridged FIR and detailed in Sec. M.2.3), is only resilient against fault injection attacks, and is also described from single-block encryptions. The AONE protocol is also resilient against fault injection attacks, when multiple blocks are encrypted.

M.2.1 Indexed Key Update (IKU)

The leakage-resilience ensures that an attacker cannot retrieve the full secret key k, assuming two hypotheses. The first one is that only computations that involve the key induce leakage. The second is that one encryption leaks much less information than the whole key.

These hypotheses indeed reflect some true facts about practical side-channel analysis. The first one is a consequence of the way nowadays electronic circuits work. They are implemented in CMOS logic, that is leakage-free in static mode. Put differently, CMOS is built on purpose not to conduct any current if no net changes. At the opposite, when there is some activity, then the nets that toggle produce a current. The aggregation of those current make up the side-channel. The second hypothesis is a consequence of this fact: in a non-invasive setup, the attacker is not able to probe an node of the circuit. Instead, through the passivation layers, she can reasonably monitor the sum of the leakage of many nets in the vicinity of her sensor. Therefore her side-channel inevitably contains some algorithmic noise, *i.e.* random activity caused by the neighbour nets. If we also take into account the finite bandwidth of the sensor and the finite accuracy of the digitalizing apparatus, it appears clearly that several measurements will be necessary, simply to get rid of all this noise.

In this context, Paul C. Kocher suggests in [246, §4] to update the key on a regular basis. Typically, an evaluator can estimate the number of key manipulations after which the full key can be recovered by an attack, because sufficient information can be garnered to overcome the effect of the random noise. Now, this number depends on the characteristic of the acquisition campaign; thus, this paper [434] suggests to use a success rate metric to estimate the strength of an attack. Using this tool, it is possible to define a number η of encryption for which the success rate is below a given threshold, say 1%. The initial proposal of Paul C. Kocher is to hash the secret key every η encryptions, and to continue the encryptions with the result of the hash $k := h(k)$. Again, after η other

key usages, the secret is replaced by its hash value. Because of the cryptographic properties of hash functions, the partial knowledge of the key before the key hash cannot be capitalized to break the new session key. The same noting can be done in the other way round: the partial information about the current key is of no help to deduce constructive information about the less iterated hashed keys. It is in this respect that this regular key update is resilient against passive attacks.

However, this key update is not appropriate to the computation with multiple parties. For the sake of illustration, we assume that the party A is a vulnerable device (say a smartcard), that communicates with many correspondents B (supposed to be secure). Now, if A has already been hashing the key a large amount of times (say $1,000,000$ times), then the next B it interacts with needs to start computing $1,000,000$ hashes on the primary secret key k before being synchronized with A. This situation seems unrealistic, and will get worse as the number of used keys increases.

Therefore, Paul C. Kocher introduces in [245] a notion of session key tree. We assume binary trees, but similar constructs can be obtained for arities greater than two. The system assumes a finite number of keys, $2^D - 1$ for some integer D. For instance, in [245], D is chosen equal to 39. In the sequel, we simply note that it is a small constant, that satisfies $D = \mathcal{O}(1)$ when the number of blocks to encrypt (n) increases. The tree is rooted by k, and constructed recursively. The two sons of a node κ are respectively $E_l(\kappa)$ and $E_r(\kappa)$, the result of the encryption of κ by one of the two reversible functions E_l or E_r. Typically, $E_{\{l,r\}}$ are encryption functions, using two different keys that are made public. Indeed, the knowledge of the transformation of one session key into another does not convey any information to a prospective attacker if the nodes value is secret. In addition, publishing the encryption keys for $E_{\{l,r\}}$ reduces the amount of shared secrets to be otherwise safely concealed in every A and B. Now, the correspondents derive their successive session keys by a depth-first left-to-right tree traversal. Moving downwards left (*resp.* downwards right, upwards left, upwards right) is achieved by the application of E_l (*resp.* E_r, E_l^{-1}, E_r^{-1}). The current key is indexed by its order $C \in [\![0, 2^D[\![$ in the tree traversal. We denote k_C the corresponding key, and note $k_0 = k$, because, by convention, the root of the tree has index $C = 0$. The figure 1 illustrates this key indexation mechanism.

The session key is agreed on between A and B by first comparing their counters C. They select the greatest of the two C for the shared index. Then, there exists an algorithm to reach any node from any other node using between 0 and $2D - 2$ calls to either of $E_{\{l,r\}}^{\{+1,-1\}}$. Once this first session key is fetched, the subsequent keys $k_{\{C+1,C+2,\cdots\}}$ are retrieved each thanks to only one call to either $E_{\{l,r\}}^{\{+1,-1\}}$, because a single displacement is required since consecutive keys are direct neighbours in the tree.

M.2.2 Fresh Re-Keying (FRK)

The previous IKU scheme has several drawbacks, for instance:

– the number of possible keys is limited to $2^D - 1$ although the key space is must larger;

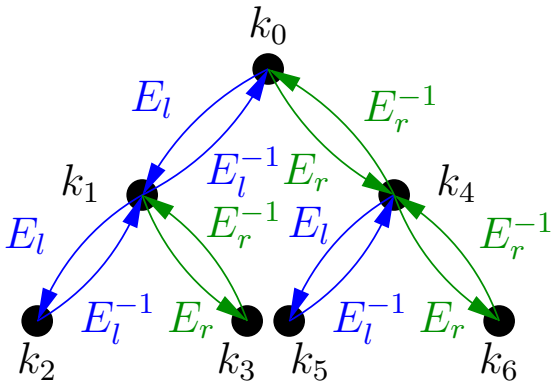


Figure 1: Illustration of IKU in a binary tree for a depth $D = 3$.

- the key agreement requires several calls to encryption/decryption functions, and this number of calls depends on the key index C;
- the correspondent A must have some tamper-proof non-volatile memory (NVM) to store C.

The paper [305] introduces an interactive session key derivation algorithm that addresses all the shortcomings of IKU. For this purpose, the authors of [305] replace the key search in a tree by a random key generation, thanks to a randomized bijection noted f. In practice, this function takes in input the root key k and a random number r, and outputs a fresh session key $k^* = f(k, r)$. A sends to B the random number r so that B is able to reproduce the cryptographic computations done by A. It is straightforward to understand that this fresh re-keying protocol can generate all possible keys, in one go, and in a state-less manner.

Nevertheless, one immediate shortcoming of this FRK arises from its interactivity. In the long run, this algorithm is more I/O consuming, since A must send a random number along with each block of encrypted data, whereas in IKU, the synchronization with C is done once at the beginning, and subsequently A and B remain in phase if they implicitly know when to increment C.

Despite of this consideration, it is worth noting that FRK can be instantiated in an implementation that enjoys a remarkably efficient session key derivation algorithm. The authors of [305] indeed underline that the security features of the protocol can be partitioned in two independent problems. On the one hand, the block cipher g keyed with $k^* = f(k, r)$ is responsible for the scheme's robustness against cryptanalytic attacks. On the other hand, the resilience against physical attacks is confined in f. There are two reasons for that. First of all, if the block cipher is invoked fewer times than the number of times η where a side-channel attack have fair chance to succeed, then k^* will definitely remain out of reach of an attacker. Now, not knowing k^*, the attacker cannot inverse $f(\cdot, r)$ if it has a good diffusion property. Second, another attack path would be to examine the leakage of f: indeed, this function is fed by an unknown secret k mixed with

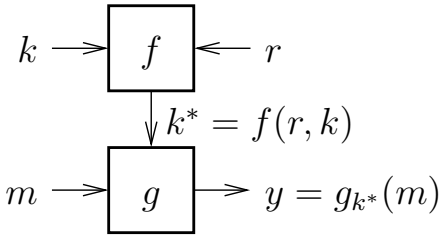


Figure 2: Illustration of a single-block encryption with FRK.

a known varying input r. This setup is canonical for a side-channel attack. Nonetheless, as defended in [305], f can be chosen to both fulfill the diffusion property and to be easily protected against side-channel attacks. In the example discussed in [305], the very nature of f (the multiplication $\times$ in a given ring) makes it easily protected by masking. Also, $f = \times$ can be implemented much more efficiently than a complete cryptographic bijection (*e.g.* $f =$ AES).

The encryption of block of data with FRK, resilient against side-channel attacks, is illustrated in Fig. 2.

M.2.3 Fault Injection Resilience (FIR)

Avoiding faults in a circuit is a difficult task, since whatever detection scheme, it is always possible, by chance, to substitute a valid data by an other valid one, that will obviously not be detected. On the contrary, the resilience approach against fault injection attacks consists in tolerating errors, but also in denying the attacker from exploiting them. We call fault injection resilient (FIR) a scheme where the attacker can neither *choose* nor *influence* the input of the encryption algorithm. This way, intuitively, the attacker has no means even to know if the result is faulted or not, and even less to know how the ciphertext is faulted. In particular, differential fault attacks are impossible since it is impossible to collect twice the ciphertext corresponding to the same plaintext, and safe errors are equally impossible since the attacker cannot distinguish between the correct and the purportedly faulted ciphertext. This FIR model would certainly deserve a more formal definition. Nonetheless, we continue with this FIR notion, defined as the input *non-forgeability* and *non-malleability*.

An example of FIR is given in [207, §3-A], and recalled in Alg. 1.

The proposed way to avoid an attacker from choosing or influencing the plaintext is to blind it with a random number r of the same size. Thus, instead of returning the encryption $g_k(x)$ of plaintext x, the algorithm returns the couple $(g_k(x \oplus r), r)$. This way of proceeding doubles the requirement for the I/O bandwidth, but is otherwise computationally equivalent to the plain unprotected g_k, if we neglect the XOR operation involved in the blinding compared to an encryption g_k.

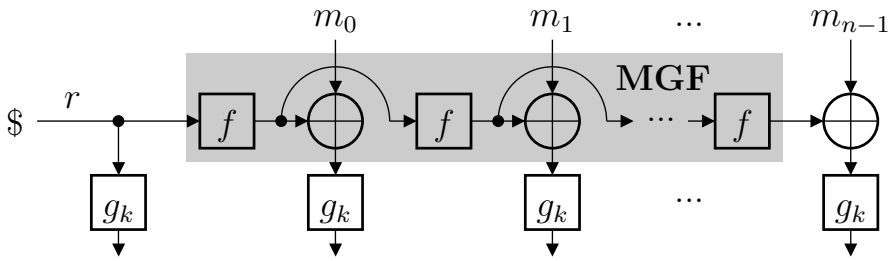


Figure 3: Resilient MGF, used as partial AONT.

M.2.4 All-Or-Nothing Encryption (AONE)

The AONE [302, 303] builds its security by denying the attacker from knowing the plaintext and the ciphertext. However, in AONE, the key is not updated: it remains k throughout the encryptions. The suggested construct is an all-or-nothing transform (AONT [384]), applied *full* on the plaintext and *partial* on the ciphertext.

We remark here that the AONT is too strong a preprocessing on the plaintext; indeed, the goal is simply to make it unpredictable to the attacker. Thus, a simple *partial* AONT, such as a lightweight mask generation function (MGF) described in Fig. 3, do achieve a randomization of the plaintext. The MGF consists in the iterated application on a random number r of a deterministic function f that can be abstracted as a random oracle (*e.g.* a lightweight encryption function with a public key or a lightweight hash function). This MGF adds one extra block r to be encrypted; this block is drawn randomly by A, and discovered upon decryption by B.

For AONE to be effective as such against passive attacks, the ciphertext shall also be blinded. However, this requires an initial secret exchange, which is a serious drawback of the approach. Indeed, sharing a secret requires an operation such as a Diffie-Hellman (DH) exchange [112]. Now, low cost devices that rely on symmetric cryptography are not equipped with the hardware to conduct DH. In addition, the DH exchange is very expensive in terms of time and code size. Therefore, this pass is really deterrent for the adoption of the AONE.

M.2.5 Synthesis about the State-of-the-Art

In the sequel, we discard FIR as such, because it does not protect against passive side-channel attacks (*i.e.* that exploit the leakage). AONE, even in our optimized version, is also discarded because of the unrealistic requirement for an initial secret sharing in addition to the key. Nonetheless, the principle of input non-forgeability and non-malleability of FIR and of our optimized version of AONE will be reused latter on in Sec. M.5 to armor IKU and FRK against active attacks.

M.3 Security Model and Security Target

The goal of this section is twofold. First of all, we define the threats that shall be taken into account by the resilience schemes we intent to compare, and also quantity them. Then, we set up the security properties we want our resilient schemes to meet.

M.3.1 Formalization of the Risks

Passive and active attacks are equally likely to be applied on an embedded system. A reasonable attacker will certainly choose the one that is the easiest. This sub-section aims at settling the actual risks that exist on insecure primitives regarding these two threats.

Regarding passive attacks, we adopt the same classifications of attacks as presented in [305]. A primitive that leaks the whole secret is said attackable by the simple power attack (SPA). Now, with such a primitive, it is very hard if not impossible to compute securely. Thus, we define primitives to be SPA-resistant if they do not disclose all the secret key by only one observation by attacker. The differential power analysis (DPA) is a statistical attack that exploits the dependence of the measurements in a binary sensitive variable. We reuse the notation η introduced previously to quantify the amount of measurements for which not enough bits of the secret have leaked to compromise it. We thus assume that an SPA-resistant primitive can be safely called η times. As we wish to build a resilient protocol without resorting to expensive *ad hoc* countermeasures, we will only consider off-the-shelf primitives. Now, any respectable intellectual property (IP) cipher can be expected to be SPA-resistant but not DPA-resistant. Thus, in the sequel, we assume that g_κ is not called more than η times with the same key κ.

Active faults can be extremely effective. When g is the AES-128, then as few as one fault can be enough to extract the 128-bit key: this has notably been shown in this differential fault analysis (DFA) [463]. As this attack is differential, it actually requires the knowledge of one correct and one (specially crafted) faulted. Thus, unless special care is taken (such as explained in the improved AONE in Sec. M.2.4), it shall not be considered secure to let the protocol encrypt two blocks with the same key. Additionally, we note that $\eta \gg 2$, *i.e.* resisting to passive leakage is easier than resisting to fault injection attacks, in terms of number of queries.

M.3.2 Common Set of Security Objectives

We consider the case of equipments sharing a common secret k and willing to use it in a block cipher, either for the purpose of authentication or (non-exclusively) for the purpose of data encryption. As already mentioned, we demand that no other secret be shared by the devices. The security of the block cipher encryptions must rely only on the resilient usage manipulation of k.

Our security objective is to compare resilience schemes that do not disclose the secret k, for a number of large transactions (much greater than η but all the same smaller than $2^D - 1$, for IKU to be admitted in the competition), and in the context of passive

Table M.1: Passive and active resilient encryption by A of a message sent by B, in IKU (*top*) and in FRK (*bottom*) protocols.

Step	**Single-block IKU**		
#1	A sends C_{A}	$\longrightarrow$	B receives C_{A}
	A receives C_{B}	$\longleftarrow$	B sends C_{B}
#2	A computes k_C as $k_C = k_{\max(C_{\mathsf{A}},C_{\mathsf{B}})}$		B computes k_C as $k_C = k_{\max(C_{\mathsf{A}},C_{\mathsf{B}})}$
#3	A receives m	$\longleftarrow$	B sends m
#4	A computes $y = g_{k_C}(m)$		
#5	A sends y	$\longrightarrow$	B receives y

Step	**Single-block FRK**		
#1	A sends r	$\longrightarrow$	B receives r
#2	A computes k^* as $k^* = f(r,k)$		B computes k^* as $k^* = f(r,k)$
#3	A receives m	$\longleftarrow$	B sends m
#4	A computes $y = g_{k^*}(m)$		
#5	A sends y	$\longrightarrow$	B receives y

and active attacks. This must be achieved using only an off-the-shelf cryptographic primitives for g, attackable as such (*i.e.* without resilience) with η traces passively and with 2 encryptions in active attacks.

The functionality is the encryption, that must be implemented by a function g that is cryptographically strong. Of course, the other functions involved in the implementation of the resilience can be *ad hoc*, and can be only resistant against active and passive attacks. Such primitives might be much less costly than cryptographic primitive. Their use can improve the efficiency (cost in terms of speed and required hardware/software) of the resilient protocol.

So far, two protocols fulfill the security requirements: IKU with single block encryptions (otherwise DFA becomes possible) and genuine FRK (with a single block also). They are sketched in Tab. M.1 for one block encryptions. In the DFA-proof setup, this protocol is merely repeated n times to request for n encryptions.

M.4 Performance Assessment

M.4.1 Authentication and Files Encryption

We consider two scenarios: symmetric authentication of devices and large files encryptions. The first case is a mere challenge-response involving the encryption of one block, whereas the second one depicts more the case of an electronic passport encrypting an identity facial picture (4 kBytes make up 512 triple-DES blocks or 256 AES-{128,192,256} blocks).

M.4.2 Performance Figures

The protocols can be appreciated according to various criteria, depending on which resource is the most limiting on the targeted device. First of all, the presence of NVM can be considered an option in extremely price-constrained devices. Or also, when migrating one application on a device with just enough NVM to accommodate the secret key k from a low protection level to a resilience-protection type, it is not an alternative to consider more NVM since the upgrade shall be done at constant resources (which is also why we do not investigate [151]). Second, protocols can be classified according to their requirement for a true random number generator (TRNG). Then, the number of exchanged messages can also be limiting, especially for contact-less devices. Eventually, the complexity in terms of execution time and processing power is a third parameter to take into account. In this respect, the cryptographic primitives are always considered most costly than *ad hoc* lightweight primitives that fulfill only one requirement, such as diffusion, while remaining at least SPA-resistant. Indeed, SPA-resistance is assumed to be a prerequisite for all the blocks involved in the execution of the protocol.

M.4.3 Results for State-of-the-Art Protocols

The IKU requires some NVM, *a minima* in a quantity equal to the key tree depth (D bits) to store the current position C. In [245], some tradeoffs are suggested, such as increasing the NVM in exchange for a greatest key localization speed. For the sake of simplicity, we decide not to explore tradeoffs between the performance figures defined in Sec. M.4.2. The only variations we will discuss (in Sec. M.5) are net optimizations without counterparts. On the contrary, FRK works without NVM. Symmetrically, IKU is deterministic, whereas FRK demands a TRNG. Regarding the amount of data to be sent in an authentication session between A and B, the key establishment procedures (step #1 in Tab. M.1) require $D + D$ bits in IKU and B bits (where B is the cipher g block size) for FRK. Taken into account the one-block reception (step #3) and reemission after encryption (step #5), one ends up with $2D + 2B$ for IKU and $3B$ for FRK. If we note $[X]$ the performance of operation X, then IKU costs between $1 \times [E]$ to $(2D-2) \times [E]$ to compute the session key on the tree, where E is one of $E_{\{l,r\}}^{\{+1,-1\}}$. Let us consider worst cases. If we neglect small comparisons and focus on operations on data and keys, then IKU costs in total $(2D-2) \times [E] + [g]$ and FRK $[f] + [g]$.

Table M.2: Summary of the performances of various resilient protocols (for 1 to n blocks).

Protocol	**I/O [bit]**	**Performance**
1-bl. IKU	$2D + 2B$	$(2D - 2)[E] + [g]$
1-bl. FRK	$3B$	$[f] + [g]$
n-bl. IKU	$2D + 2Bn$	$(2D - 3 + n)[E] + n[g]$
n-bl. FRK	$3Bn$	$n \cdot ([f] + [g])$
n-bl. IKU+	$2D+(1+\frac{\eta}{\eta-1})nB$	$(2D-3+\frac{n}{\eta-1})[E]+n(\frac{\eta}{\eta-1}[g]+[f])$
n-bl. FRK+	$(\frac{2\eta}{\eta-1})nB$	$\frac{n}{\eta-1}[f] + n(\frac{\eta}{\eta-1}[g] + [f])$
n-bl. IKU*	$2D + 2Bn$	$(2D - 3 + n)[f] + n[g]$
n-bl. IKU+*	$2D+(1+\frac{\eta}{\eta-1})nB$	$(2D-3+\frac{n}{\eta-1})[f]+n(\frac{\eta}{\eta-1}[g]+[f])$
n-bl. FRK+H	$B + (1 + \frac{\eta}{\eta-1})nB$	$\frac{n}{\eta-1}[f] + n(\frac{\eta}{\eta-1}[g] + [f])$
• IKU & IKU* require ..... NVM but no TRNG; • IKU+ & IKU+* require ..... both NVM and TRNG; • FRK, FRK+ & FRK+H require ... TRNG but no NVM.		

Now, if instead of the authentication case, we focus on the large file encryption case, then n applications of g shall be considered. In IKU, the exchange of the positions in the tree is done only for the first block; afterwards, we can assume A and B implicitly know that the next key can be found at the next position, with one application of $E_{\{l,r\}}^{\{+1,-1\}}$. As far as FRK is concerned, encrypting multiple blocks consists in replaying the same single block protocol with a new random r each time. Thus, for long messages ($n \gg 1$), IKU becomes more bandwidth-efficient than FRK, whereas FRK keeps its performance advantage (since $[f] < [E] = [g]$, because f is lightweight whereas E and g are full-fledged cryptographic block ciphers). These results are summarized in the four first rows of Tab. M.2.

M.5 Improvement of the State-of-the-Art in the Encryption of Large Files Scenario

M.5.1 Armoring IKU and FRK on $n > 1$ Blocks against Fault Attacks: IKU+ and FRK+

We present IKU+ and FRK+, that are multi-block versions of IKU and FRK. The two latter protocols could not use multiple blocks because of fault attacks. Now, if a MGF is applied on the plaintext, then the plaintext actually becomes non-forgeable and non-malleable, thus fault injection resilient. Therefore, one session key can be reused for

Figure 4: The improved FRK+H scheme requires only one initial interaction to derive the first ephemeral key k_1^*. Subsequent keys $k_{i>1}^*$ are deduced from k_{i-1}^* by lightweight hashing with h.

up to η blocks — beyond, DPA is a concern.

To simplify the comparisons, we assume that n is large ($n > \eta$), and multiple of η. With one session key, we can safely encrypt η blocks. Now, in η blocks, one is reserved to encrypt the random nonce r involved in MGF (recall Fig. 3). Thus, $\frac{n}{\eta-1}$ session keys are required. In IKU+, the first key look-up still costs $(2D-2)\cdot[E]$, and the other keys continue to cost $[E]$ each, however their number is reduced from $n-1$ down to $\frac{n}{\eta-1}-1$. The introduction of the MGF adds $\eta-1$ operations "f" per session key, hence n calls to f. However, the bandwidth is unchanged. FRK+ profits from the MGF both in terms of bandwidth and performance, as shown in Tab. M.2.

M.5.2 Improving IKU with Lightweight Key-Update: IKU+*

FRK makes use of a lightweight primitive f instead of a cryptographic-grade primitive (E) to derive the session keys. The same could be done for IKU. We call IKU* (*resp.* IKU+*) the version of IKU (*resp.* IKU+) where E is replaced by a lightweight ersatz (of similar complexity as the f in [305]) that does not alter the security level since they manipulate unknown quantities.

Related keys attacks have been reported recently on AES-192 and AES-256. They are cryptanalytic attacks that require two or more encryptions with related keys, *i.e.* differing by only a few bits. To be immune from these attacks, one must make sure that the update function is not two trivial. However, a primitive such as $f = *$ certainly has an high enough dispersion to prevent such related keys to be produced frequently.

M.5.3 Synchronous Session Keys Update by Iterative Hashing: FRK+H

The first session key must be agreed on, deterministically as for IKU and probabilistically as for FRK. But the next session keys can be obtained from another protocol, such as iterative hashing of the first session key (as suggested in [246, §4]). For FRK+, this allows to still improve on the I/O bandwidth, without altering the resilience property. This scheme is called FRK+H, and its performance is given in Tab. M.2. It is illustrated in Fig. 4.

For IKU+*, such a transformation would trade a lightweight key update with f on the tree with a lightweight hash, we assume to have the same cost. Therefore, we do not consider it (because it does not change the performance of IKU+*).

At this stage, we have optimized as much as possible IKU and FRK: the best protocols for large files encryption are IKU+* and FRK+H. The result is that

- in term of I/O bandwidth, the requirements are the same when $n \to \infty$ (namely $(1 + \frac{\eta}{\eta-1})nB$, up to a negligible constant);
- Computation-wise, both require exactly $n\frac{\eta}{\eta-1}$ calls to g and about $n(1+\frac{1}{\eta-1})$ calls to a lightweight f.

Thus, the differences observed for short messages (authentication case) tend to fade and asymptotically, the optimized protocols are equivalent when dealing with the encryption of large messages.

M.5.4 Other Implementation-Dependant Considerations to Tune the Resilience Schemes

Each time the protocol reduces the number of key updates, as in Sec. M.5.1, the key schedule step of g is saved. Now, this step is both timing consuming (especially on AES) and source of an extra leakage.

M.6 Conclusions and Perspectives

We have investigated resilient computation schemes for both hardware and software implementations. Our study shows that, amongst the known schemes, those based on a regular key update are effective. Two solutions (IKU and FRK) exist, depending whether the key sequence is deterministic or probabilistic. We show that using state-of-the-art protocols, FRK is always more efficient for single-block encryption, whereas IKU is always less I/O consuming for large files encryptions.

However, as such, IKU and FRK can only use one-block payload exchange, because of fault injection analyses. By blinding the input, we show that multi-blocks can be used, which improve the performance (leading to the new protocols we nickname IKU+ and FRK+).

In addition, we propose another series of improvements, after which the two schemes tend to be equivalent (when the number of blocks to process n becomes larger and larger). Thus, we confirm that, for an identical security objective, the use of lightweight primitives [360] in conjunction with cryptographic primitives can indeed enhance the efficiency of the protocol. As a perspective, we expect interesting researches on this topic to continue make the cost of resilience-based protections more acceptable.

We also underline that eventually, the last optimisation level will be done by instantiating the most efficient primitives given the data & key bitwidths. Therefore, we point out the requirement for "stretchable" lightweight primitives (in terms of data bitwidth and latency) operating as diffusion primitives between real cryptographic-grade primitives making up the functional skeleton of the protocol.

Bibliography

[1] Common Criteria (*aka* CC) for Information Technology Security Evaluation (standard ISO/IEC 15408). Website: http://www.commoncriteriaportal.org/. 10, 210, 327

[2] Moulay Abdelaziz El Aabid, Sylvain Guilley, and Philippe Hoogvorst. Template Attacks with a Power Model. Cryptology ePrint Archive, Report 2007/443, December 2007. http://eprint.iacr.org/2007/443/. 58, 195

[3] Moulay Abdelaziz El Aabid, Oliver Meynard, Sylvain Guilley, and Jean-Luc Danger. Combined Side-Channel Attacks. In *WISA*, volume 6513 of *LNCS*, pages 175–190. Springer, August 24-26 2010. Jeju Island, Korea. DOI: 10.1007/978-3-642-17955-6_13. vii, 58, 233

[4] Michel Agoyan, Jean-Max Dutertre, David Naccache, Bruno Robisson, and Assia Tria. When Clocks Fail: On Critical Paths and Clock Faults. In *CARDIS*, volume 6035 of *Lecture Notes in Computer Science*, pages 182–193. Springer, April 14-16 2010. Passau, Germany. 306

[5] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. Multi-channel Attacks. In *CHES*, volume 2779 of *LNCS*, pages 2–16. Springer, September 8-10 2003. Cologne, Germany. 235

[6] Mehdi-Laurent Akkar and Christophe Giraud. An Implementation of DES and AES Secure against Some Attacks. In LNCS, editor, *Proceedings of CHES'01*, volume 2162 of *LNCS*, pages 309–318. Springer, May 2001. Paris, France. 12, 80, 85, 146, 192

[7] Mohamed W. Allam and Mohamed I. Elmasry. Dynamic current mode logic (DyCML), a new low-power/high-performance logic family. In *IEEE Custom Integrated Circuits Conference (CICC)*, pages 421–424, 2000. DOI: 10.1109/CICC.2000.852699. 125

[8] VSI Alliance. On-Chip Bus Development Working Group. Virtual Component Interface (VCI) Standard Version 2 (OCB 2 2.0), April 2001. http://www.vsia.org/. 123, 150

[9] Frédéric Amiel, Benoît Feix, and Karine Villegas. Power analysis for secret recovering and reverse engineering of public key algorithms. In *Selected Areas in Cryptography*, pages 110–125, August 16 & 17 2007. Ottawa, Ontario, Canada. 250, 252

[10] Frédéric Amiel, Karine Villegas, Benoît Feix, and Louis Marcel. Passive and Active Combined Attacks: Combining Fault Attacks and Side Channel Analysis. In *FDTC*, pages 92–102. IEEE Computer Society, 10 September 2007. Vienna, Austria. 235, 293

[11] Ross J. Anderson and Markus G. Kuhn. Tamper Resistance – a Cautionary Note. In *In Proceedings of the Second USENIX Workshop ON Electronic Commerce*, pages 1–11, November 18-21 1996. Oakland, California. ISBN 1-880446-83-9. 310

[12] Ross J. Anderson and Markus G. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In *Security Protocols Workshop*, volume 1361 of *Lecture Notes in Computer Science*, pages 125–136. Springer, April 7-9 1997. Paris, France. 251, 306

[13] Cédric Archambeau, Éric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In *CHES*, volume 4249 of *LNCS*, pages 1–14. Springer, October 10-13 2006. Yokohama, Japan. 28, 120, 139, 143, 180, 183, 242

[14] Yuichi Baba, Atsushi Miyamoto, Naofumi Homma, and Takafumi Aoki. Multiple-Valued Constant-Power Adder for Cryptographic Processors. In *ISMVL*, pages 239–244. IEEE Computer Society, May 21-23 2009. Naha, Okinawaw, Japan. 315

[15] Stéphane Badel, Erdem Guleyupoglu, Ozgur Inac, Anna Pena Martinez, Paolo Vietti, Frank K. Gürkaynak, and Yusuf Leblebici. A Generic Standard Cell Design Methodology for Differential Circuit Styles. In *DATE*, pages 843–848. IEEE, 2008. 93

[16] Benoît Badrignans, Jean-Luc Danger, Viktor Fischer, Guy Gogniat, and Lionel Torres. *Security Trends for FPGAS – From Secured to Secure Reconfigurable Systems*. Springer, June 20 2011. DOI: 10.1007/978-94-007-1338-3. 2

[17] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstal, and Claire Whelan. The Sorcerer's Apprentice Guide to Fault Attacks. *Proceedings of the IEEE*, 94(2):370–382, 2006. DOI: 10.1109/JPROC.2005.862424. 271, 312

[18] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. The Sorcerer's Apprentice Guide to Fault Attacks. *Proceedings of the IEEE*, 94(2):370 –382, February 2006. 2

[19] Alessandro Barenghi, Guido Bertoni, Emanuele Parrinello, and Gerardo Pelosi. Low voltage fault attacks on the RSA cryptosystem. In *FDTC*, pages 23–31. IEEE Computer Society, September 6th 2009. Lausanne, Switzerland. DOI: 10.1109/FDTC.2009.30. 306

[20] Alessandro Barenghi, Guido Bertoni Luca Breveglieri, Mauro Pellicioli, and Gerardo Pelosi. Low Voltage Fault Attacks to AES. In *HOST (Hardware Oriented Security and Trust)*. IEEE Computer Society, June 13-14 2010. Anaheim Convention Center, CA, USA. DOI: 10.1109/HST.2010.5513121. 306

[21] Lejla Batina, Benedikt Gierlichs, and Kerstin Lemke-Rust. Comparative Evaluation of Rank Correlation Based DPA on an AES Prototype Chip. In *ISC*, volume 5222 of *Lecture Notes in Computer Science*, pages 341–354. Springer, September 15-18 2008. Taipei, Taiwan. 77

[22] Lejla Batina, Benedikt Gierlichs, and Kerstin Lemke-Rust. Differential Cluster Analysis. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems – CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 112–127, Lausanne, Switzerland, 2009. Springer-Verlag. 30, 77

[23] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual Information Analysis: a Comprehensive Study. *J. Cryptology*, 24(2):269–291, 2011. 209

[24] Friedrich Beck. *Integrated Circuit Failure Analysis: A Guide to Preparation Techniques*. Wiley, January 1998. ISBN-10: 0471974013; ISBN-13: 978-0471974017; 190 pages. 3, 306

[25] Olivier Benoît and Thomas Peyrin. Side-Channel Analysis of Six SHA-3 Candidates. In *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 140–157. Springer, August 17-20 2010. Santa Barbara, CA, USA. 11, 12

[26] Guido Bertoni, Luca Breveglieri, Israel Koren, Paolo Maistri, and Vincenzo Piuri. Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard. *IEEE Trans. Computers*, 52(4):492–505, 2003. 270

[27] Guido Bertoni, Marco Macchetti, Luca Negri, and Pasqualina Fragneto. Power-Efficient ASIC Synthesis of Cryptographic S-Boxes. In *GLSVLSI '04: Proc. of the 14th ACM Great Lakes symposium on VLSI*, pages 277–281. ACM, April 2004. Boston, MA, USA. 158

[28] Régis Bevan and Erik Knudsen. Ways to Enhance Differential Power Analysis. In *ICISC*, volume 2587 of *Lecture Notes in Computer Science*, pages 327–342. Springer, November 28-29 2002. Seoul, Korea. 77

[29] Taha Beyrouthy, Laurent Fesquet, Alin Razafindraibe, Sumanta Chaudhuri, Sylvain Guilley, Philippe Hoogvorst, Jean-Luc Danger, and Marc Renaudin. A Secure Programmable Architecture with a Dedicated Tech-mapping Algorithm: Application to a Crypto-Processor. In *DCIS*, Grenoble, France, nov 2008. IEEE. 58

[30] Taha Beyrouthy, Alin Razafindraibe, Laurent Fesquet, Marc Renaudin, Sumanta Chaudhuri, Sylvain Guilley, Philippe Hoogvorst, and Jean-Luc Danger. A Novel Asynchronous e-FPGA Architecture for Security Applications. pages 369–372. IEEE, Dec 2007. FPT'07, Kokurakita, Kitakyushu, Japan. 58

[31] Shivam Bhasin, Taoufik Chouta, Guillaume Duc, Jean-Luc Danger, Aziz El Aabid, Florent Flament, Philippe Hoogvorst, Tarik Graba, Sylvain Guilley, Houssem Maghr'ebi, Olivier Meynard, Maxime Nassar, Renaud Pacalet, Laurent Sauvage, Nidhal Selmane, and Youssef Souissi. Combined countermeasures against perturbation & observation attacks. In *PASTIS (PAca Security Trends In embedded Security)*, Gardanne (École des Mines de Saint-Étienne), France, June 16-17 2010. http://www.secure-ic.com/PDF/pastis_2010.pdf. 58, 315

[32] Shivam Bhasin, Taoufik Chouta, Guillaume Duc, Jean-Luc Danger, Aziz Elaabid, Florent Flament, Philippe Hoogvorst, Tarik Graba, Sylvain Guilley, Houssem Maghrebi, Olivier Meynard, Maxime Nassar, Renaud Pacalet, Laurent Sauvage, Nidhal Selmane, and Youssef Souissi. DPA et Dérivées : Attaques et Contremesures, March 31st 2010. GDR SoC-SiP, Paris, France. http://www.lirmm.fr/journees_securite/material/j2/Guilley.pdf. 58

[33] Shivam Bhasin, Jean-Luc Danger, Florent Flament, Tarik Graba, Sylvain Guilley, Yves Mathieu, Maxime Nassar, Laurent Sauvage, and Nidhal Selmane. Combined SCA and DFA Countermeasures Integrable in a FPGA Design Flow. In *ReConFig*, pages 213–218. IEEE Computer Society, December 9–11 2009. Cancún, Quintana Roo, México, DOI: 10.1109/ReConFig.2009.50, http://hal.archives-ouvertes.fr/hal-00411843/en/. viii, 39, 58, 290, 291, 317, 322, 323

[34] Shivam Bhasin, Jean-Luc Danger, Florent Flament, Tarik Graba, Sylvain Guilley, Yves Mathieu, Maxime Nassar, Laurent Sauvage, and Nidhal Selmane. Combined SCA and DFA Countermeasures Integrable in a FPGA Design Flow. In *ReConFig*, pages 213–218. IEEE Computer Society, December 9–11 2009. Cancún, Quintana Roo, México, DOI: 10.1109/ReConFig.2009.50. 84

[35] Shivam Bhasin, Jean-Luc Danger, Tarik Graba, and Sylvain Guilley. How to design BCDL Logic with the best Trade-off between Complexity and Robustness. In *CryptArchi*, Bochum, Germany, June 15–18 2011. Bochum, Germany; (abstract). 58

[36] Shivam Bhasin, Sylvain Guilley, and Jean-Luc Danger. From Cryptography to Hardware: Analyzing Embedded Xilinx BRAM for Cryptographic Applications. In *HASP*, pages 1–8. IEEE, December 2nd 2012. Vancouver, British Columbia, Canada. DOI: 10.1109/MICROW.2012.11 . 58

[37] Shivam Bhasin, Sylvain Guilley, Florent Flament, Nidhal Selmane, and Jean-Luc Danger. Countering Early Evaluation: An Approach Towards Robust Dual-Rail Precharge Logic. In *WESS*, pages 6:1–6:8. ACM, October 24-28 2010. Scottsdale, Arizona, USA. DOI: 10.1145/1873548.1873554. 34, 58, 323

[38] Shivam Bhasin, Sylvain Guilley, Annelie Heuser, and Jean-Luc Danger. From cryptography to hardware: analyzing and protecting embedded Xilinx BRAM for cryptographic applications. *Journal of Cryptographic Engineering*, 3(1), 2013. 58

[39] Shivam Bhasin, Sylvain Guilley, Laurent Sauvage, and Jean-Luc Danger. Unrolling Cryptographic Circuits: A Simple Countermeasure Against Side-Channel Attacks. In *RSA Cryptographers' Track, CT-RSA*, volume 5985 of *LNCS*, pages 195–207. Springer, March 1-5 2010. San Francisco, CA, USA. DOI: 10.1007/978-3-642-11925-5_14. vi, 58, 191

[40] Shivam Bhasin, Sylvain Guilley, Youssef Souissi, and Jean-Luc Danger. Efficient FPGA Implementation of dual-rail countermeasures using Stochastic Models, September 26-27 2011. Non-Invasive Attack Testing Workshop (NIAT 2011), co-organized by NIST & AIST. Todai-ji Cultural Center, Nara, Japan. (PDF). 34, 58

[41] Shivam Bhasin, Sylvain Guilley, Youssef Souissi, Tarik Graba, and Jean-Luc Danger. DPL Implementations in FPGA using Embedded BRAM. In *TrustED, First International Workshop on Trustworthy Embedded*, September 15-16 2011. Leuven, Belgium. 38, 58

[42] Shivam Bhasin, Sylvain Guilley, Youssef Souissi, Tarik Graba, and Jean-Luc Danger. Efficient Dual-Rail Implementations in FPGA using Block RAMs. In *ReConFig*, pages 261–267. IEEE Computer Society, November 30 – December 2 2011. Cancún, Quintana Roo, México. DOI: 10.1109/ReConFig.2011.32. 19, 34, 58

[43] Shivam Bhasin, Nidhal Selmane, Sylvain Guilley, and Jean-Luc Danger. Security Evaluation of Different AES Implementations Against Practical Setup Time Violation Attacks in FPGAs. In

HOST (Hardware Oriented Security and Trust), pages 15–21. IEEE Computer Society, July 27th 2009. DOI: 10.1109/HST.2009.5225057; In conjunction with DAC-2009, Moscone Center, San Francisco, CA, USA. 58

[44] Eli Biham and Adi Shamir. Differential Cryptanalysis of the Full 16-Round DES. In Ernest F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 487–496. Springer, 1992. 1

[45] Eli Biham and Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In *CRYPTO*, volume 1294 of *LNCS*, pages 513–525. Springer, August 1997. Santa Barbara, California, USA. DOI: 10.1007/BFb0052259. 192, 270, 292, 312, 317, 318

[46] Alex Biryukov, Adi Shamir, and David Wagner. Real Time Cryptanalysis of A5/1 on a PC. In Bruce Schneier, editor, *FSE*, volume 1978 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2000. 253

[47] Johannes Blömer and Jean-Pierre Seifert. Fault based cryptanalysis of the Advanced Encryption Standard. In Springer, editor, *Financial Cryptography*, volume 2742 of *LNCS*, pages 162–181, 2003. 270

[48] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. Cryptology ePrint Archive, Report 2011/449, 2011. http://eprint.iacr.org/. 1

[49] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 37–51, Berlin, Heidelberg, 1997. Springer-Verlag. 306, 311, 315

[50] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of Eliminating Errors in Cryptographic Computations. *Journal of Cryptology*, 14(2):101–119, 2001. 270

[51] Johan Borst. *Block ciphers: Design, Analysis and Side-Channel Analysis*. PhD thesis, K.U.L., September 2001. Leuven, Belgium. https://www.cosic.esat.kuleuven.be/publications/thesis-13.pdf. 3

[52] Arnaud Boscher and Helena Handschuh. Masking Does Not Protect Against Differential Fault Attacks. In *FDTC, 5th Workshop on Fault Detection and Tolerance in Cryptography, IEEE-CS*, pages 35–40, aug 2008. DOI: 10.1109/FDTC.2008.12, Washington, DC, USA. 9, 289, 294

[53] Arnaud Boscher, Helena Handschuh, and Elena Trichina. Blinded Fault Resistant Exponentiation Revisited. In *FDTC*, pages 3–9. IEEE Computer Society, September 6 2009. Lausanne, Switzerland. 311

[54] Arnaud Boscher, Robert Naciri, and Emmanuel Prouff. CRT RSA Algorithm Protected Against Fault Attacks. In *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems*, volume 4462 of *LNCS*, pages 229–243. Springer, May 9-11 2007. Heraklion, Crete, Greece. 311

[55] Ghislain Freddy Bouesse, Marc Renaudin, Bruno Robisson, Édith Beigné, Pierre-Yvan Liardet, Solenn Prevosto, and Jacques Sonzogni. DPA on Quasi Delay Insensitive Asynchronous Circuits: Concrete Results. In *XIX Conference on Design of Circuits and Integrated Systems, Proceedings of DCIS'04)*, 24–26 Nov 2004. Bordeaux, France (PDF). 104, 146

[56] Sébastien Briais, Stéphane Caron, Jean-Michel Cioranesco, Jean-Luc Danger, Sylvain Guilley, Jacques-Henri Jourdan, Arthur Milchior, David Naccache, and Thibault Porteboeuf. 3D Hardware Canaries. In *CHES*, September 9-12 2012. Leuven, Belgium. Full version [57]. 58

[57] Sébastien Briais, Stéphane Caron, Jean-Michel Cioranesco, Jean-Luc Danger, Sylvain Guilley, Jacques-Henri Jourdan, Arthur Milchior, David Naccache, and Thibault Porteboeuf. 3D Hardware Canaries. Cryptology ePrint Archive, Report 2012/324, 2012. http://eprint.iacr.org/2012/324/. 346

[58] Sébastien Briais, Jean-Michel Cioranesco, Jean-Luc Danger, Sylvain Guilley, David Naccache, and Thibault Porteboeuf. Random active shield. In Guido Bertoni and Benedikt Gierlichs, editors, *FDTC*, pages 103–113. IEEE, 2012. 58

[59] Sébastien Briais, Sylvain Guilley, and Jean-Luc Danger. A formal study of two physical countermeasures against side channel attacks. PROOFS workshop – Cryptology ePrint Archive, Report 2012/430, September 13 2012. http://www.proofs-workshop.org/, Leuven, Belgium. http://eprint.iacr.org/2012/430. 58

[60] Éric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, August 11–13 2004. Cambridge, MA, USA. 16, 77, 81, 120, 138, 156, 186, 193, 198, 209, 234, 254, 257, 258, 260, 270

[61] Marco Bucci, Luca Giancane, Raimondo Luzzi, and Alessandro Trifiletti. Three-Phase Dual-Rail Pre-charge Logic. In *CHES*, volume 4249 of *LNCS*, pages 232–241. Springer, October 10-13 2006. Yokohama, Japan. DOI: 10.1007/11894063. 125

[62] A. Bystrov and J.P. Murphy. On-line IDDQ testing of security circuits, 2004. School of Electrical, Electronic & Computer Engineering, University of Newcastle upon Tyne. 106

[63] Cecile Canovas and Jessy Clediere. What do S-boxes Say in Differential Side Channel Attacks? Cryptology ePrint Archive, Report 2005/311, 2005. http://eprint.iacr.org/. 203, 204

[64] Claude Carlet. On Highly Nonlinear S-Boxes and Their Inability to Thwart DPA Attacks. In *INDOCRYPT*, volume 3797 of *LNCS*, pages 49–62. Springer, december 2005. Bangalore, India. (PDF on SpringerLink; Complete version on IACR ePrint). 165

[65] Claude Carlet. Boolean Functions for Cryptography and Error Correcting Codes: Chapter of the monography Boolean Models and Methods in Mathematics, Computer Science, and Engineering. pages 257–397. Cambridge University Press, Y. Crama and P. Hammer eds, 2010. Preliminary version available at http://www.math.univ-paris13.fr/~carlet/chap-fcts-Bool-corr.pdf. 214, 216

[66] Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage Squeezing of Order Two. In *INDOCRYPT*, volume 7668 of *LNCS*, pages 120–139. Springer, December 9-12 2012. Kolkata, India. 58

[67] Claude Carlet, Philippe Gaborit, Jon-Lark Kim, and Patrick Solé. A new class of codes for Boolean masking of cryptographic computations, October 6 2011. http://arxiv.org/abs/1110.1193. To appear in IEEE Transactions on Information Theory. 16

[68] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *CRYPTO*, volume 1666 of *LNCS*. Springer, August 15-19 1999. Santa Barbara, CA, USA. ISBN: 3-540-66347-9. 246, 306

[69] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer, August 2002. San Francisco Bay (Redwood City), USA. 11, 77, 120, 139, 200, 210, 234, 258

[70] Sumanta Chaudhuri, Jean-Luc Danger, and Sylvain Guilley. Efficient Modeling and Floorplanning of Embedded-FPGA Fabric. In *FPL*, pages 665–669. IEEE, Aug 27-29 2007. Amsterdam, Netherlands. ISBN: 1-4244-1060-6. DOI: 10.1109/FPL.2007.4380741. 58

[71] Sumanta Chaudhuri, Jean-Luc Danger, Sylvain Guilley, and Philippe Hoogvorst. FASE: An Open Run-Time Reconfigurable FPGA Architecture for Tamper-Resistant and Secure Embedded Systems. In *IEEE 3rd international conference on reconfigurable computing and FPGAs (Reconfig)*, pages 1–9, San Luis Potosí, México, Sep 2006. IEEE. DOI: 10.1109/RECONF.2006.307752. 58, 293

[72] Sumanta Chaudhuri, Jean-Luc Danger, Philippe Hoogvorst, and Sylvain GUILLEY. Efficient Tiling Patterns for Reconfigurable Gate Arrays. In *SLIP'08*, pages 11–18, Newcastle University, UK, apr 2008. 58

[73] Sumanta Chaudhuri, Jean-Luc Danger, Philippe Hoogvorst, and Sylvain GUILLEY. Efficient Tiling Patterns for Reconfigurable Gate Arrays (poster session 1). In *FPGA*, page 257, Monterey, California, USA, feb 2008. 58

[74] Sumanta Chaudhuri and Sylvain Guilley. Side-Channel Oscilloscope. *CoRR*, abs/1103.1824, March 2011. 58

[75] Sumanta Chaudhuri, Sylvain Guilley, Florent Flament, Philippe Hoogvorst, and Jean-Luc Danger. An 8x8 Run-Time Reconfigurable FPGA Embedded in a SoC. In *DAC*, pages 120–125, Anaheim, CA, USA, jun 2008. ACM/IEEE. 58

[76] Sumanta Chaudhuri, Sylvain Guilley, Philippe Hoogvorst, Jean-Luc Danger, Taha Beyrouthy, Alin Razafindraibe, Laurent Fesquet, and Marc Renaudin. Physical Design of FPGA Interconnect to Prevent Information Leakage. In *ARC (Applied Reconfigurable Computing), Proceedings in LNCS Springer-Verlag Berlin Heidelberg*, volume 4943, pages 87–98, London, UK, mar 2008. 58

[77] Sumanta Chaudhuri, Sylvain Guilley, Philippe Hoogvorst, Jean-Luc Danger, Taha Beyrouthy, Alin Razafindraibe, Laurent Fesquet, and Marc Renaudin. A Secure Asynchronous FPGA Architecture, Experimental Results and Some Debug Feedback. *CoRR*, abs/1103.1360, March 2011. 58

[78] Chien-Ning Chen and Sung-Ming Yen. Differential fault analysis on AES key schedule and some countermeasures. In Springer, editor, *Information Security and Privacy*, volume 2727 of *LNCS*, pages 118–129, 2003. 271

[79] Zhimin Chen and Yujie Zhou. Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage. In *CHES*, volume 4249 of *LNCS*, pages 242–254. Springer, October 10-13 2006. Yokohama, Japan, http://dx.doi.org/10.1007/11894063_20. 295, 323, 324

[80] Zouha Cherif, Florent Flament, Jean-Luc Danger, Shivam Bhasin, Sylvain Guilley, and Hervé Chabanne. Evaluation of White-Box and Grey-Box Noekeon Implementations in FPGA. In Prasanna et al. [361], pages 310–315. 58

[81] Benoît Chevallier-Mames, Mathieu Ciet, and Marc Joye. Low-Cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity. *IEEE Trans. Computers*, 53(6):760–768, 2004. 79

[82] Zouha Chérif, Jean-Luc Danger, Sylvain Guilley, and Lilian Bossuet. An Easy-to-Design PUF based on a single oscillator: the Loop PUF. In *DSD*, September 5-8 2012. Çeşme, Izmir, Turkey; (Online PDF). 58

[83] Christophe Clavier. Side Channel Analysis for Reverse Engineering (SCARE), February 19 2004. http://eprint.iacr.org/2004/049/, 2004. Cryptology ePrint Archive: Report 2004/049. 252

[84] Christophe Clavier. An Improved SCARE Cryptanalysis Against a Secret A3/A8 GSM Algorithm. In *ICISS*, volume 4812 of *LNCS*, pages 143–155. Springer, 2007. Delhi, India. DOI: 10.1007/978-3-540-77086-2_11. 250, 252

[85] Christophe Clavier. *De la Sécurité des Cryptosystèmes Embarqués*. PhD thesis, (french). Université de Versailles Saint-Quentin-en-Yvelines, November 23 2007. 284

[86] Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Çetin Kaya Koç and Christof Paar, editors, *CHES*, volume 1965 of *Lecture Notes in Computer Science*, pages 252–263. Springer, 2000. 78

[87] Christophe Clavier, Benoît Feix, Georges Gagnerot, and Mylène Roussellet. Passive and Active Combined Attacks on AES. In *FDTC*, pages 10–18. IEEE Computer Society, 21 August 2010. Santa Barbara, CA, USA. DOI: 10.1109/FDTC.2010.17. 235

[88] "Multi-Project Wafers website", http://cmp.imag.fr/, 2007. 121

[89] Jean-Sébastien Coron, Paul C. Kocher, and David Naccache. Statistics and Secret Leakage. In *Financial Cryptography*, volume 1962 of *Lecture Notes in Computer Science*, pages 157–173. Springer, February 20-24 2000. Anguilla, British West Indies. 75, 234

[90] Jean-Sébastien Coron and Avradip Mandal. PSS Is Secure against Random Fault Attacks. In *ASIACRYPT*, volume 5912 of *LNCS*, pages 653–666. Springer, December 6-10 2009. Tōkyō, Japan. 10, 315, 330

[91] Common Criteria. Application of Attack Potential to Smartcards, Mandatory Technical Document, Version 2.7, Revision 1, CCDB-2009-03-001, March 2009. http://www.commoncriteriaportal.org/files/supdocs/CCDB-2009-03-001.pdf. 210

[92] Jean-Luc Danger, Guillaume Duc, Sylvain Guilley, and Laurent Sauvage. Education and open benchmarking on side-channel analysis with the DPA contests, September 26-27 2011. Non-Invasive Attack Testing Workshop (NIAT 2011), co-organized by NIST & AIST. Todai-ji Cultural Center, Nara, Japan. (PDF). 58

[93] Jean-Luc Danger and Sylvain Guilley. Circuit de cryptographie programmable – Logique BCDL (Balanced Cell-based Differential Logic), 25 Mars 2008. Brevet Français FR08/51904, assigné à l'Institut TELECOM; WO/2009/118264. 58, 323

[94] Jean-Luc Danger and Sylvain Guilley. Protection des modules de cryptographie contre les attaques en observation d'ordre élevé sur les implémentations à base de masquage, 20 Janvier 2009. Brevet Français FR09/50341, assigné à l'Institut TELECOM. 42, 58

[95] Jean-Luc Danger, Sylvain Guilley, Lyonel Barthe, and Pascal Benoît. *Chapter 4, "Countermeasures Against Physical Attacks in FPGAs", in "Security Trends for FPGAS – From Secured to Secure Reconfigurable Systems"*. Springer, June 20 2011. DOI: 10.1007/978-94-007-1338-3. 58

[96] Jean-Luc Danger, Sylvain Guilley, Shivam Bhasin, and Maxime Nassar. Overview of Dual Rail with Precharge Logic Styles to Thwart Implementation-Level Attacks on Hardware Cryptoprocessors. In *SCS*, IEEE, pages 1–8, November 6–8 2009. Jerba, Tunisia. Complete version online: `http://hal.archives-ouvertes.fr/hal-00431261/en/`. DOI: 10.1109/ICSCS.2009.5412599. 80

[97] Jean-Luc Danger, Sylvain Guilley, Shivam Bhasin, and Maxime Nassar. Overview of Dual Rail with Precharge Logic Styles to Thwart Implementation-Level Attacks on Hardware Cryptoprocessors, — *New Attacks and Improved Counter-Measures* —. In *SCS*, IEEE, pages 1–8, November 6–8 2009. Jerba, Tunisia. DOI: 10.1109/ICSCS.2009.5412599. 32, 58, 157, 318

[98] Jean-Luc Danger, Sylvain Guilley, and Florent Flament. Détection de faute dans un crypto-processeur protégé contre la DPA par logique différentielle, 12 Août 2008. Brevet Français FR08/55537, assigné à l'Institut TELECOM. 58, 326

[99] Jean-Luc Danger, Sylvain Guilley, and Philippe Hoogvorst. Fast True Random Generator in FPGAs. pages 506–509, Aug 2007. IEEE MWSCAS/NEWCAS'07, Montréal, Canada. 58

[100] Jean-Luc Danger, Sylvain Guilley, and Philippe Hoogvorst. Procédé de test de circuits de cryptographie et circuit de cryptographie sécurisé apte à être testé, 25 Février 2008. Brevet Français FR08/51184, assigné à l'Institut TELECOM, ayant reçu une autorisation de divulgation par la DGA; WO/2009/106428. 58

[101] Jean-Luc Danger, Sylvain Guilley, and Philippe Hoogvorst. High Speed True Random Number Generator based on Open Loop Structures in FPGAs. *Microelectronics Journal*, 40(11):1650–1656, November 2009. DOI: 10.1016/j.mejo.2009.02.004. 58

[102] Jean-Luc Danger, Sylvain Guilley, and Philippe Hoogvorst. Logiciel "`OpenLoop-TRNG`", March 16 2010.
Dépôt auprès de l'APP numéro : IDDN.FR.001.110004.000.S.P.2010.000.20000. 58

[103] Jean-Luc Danger, Sylvain Guilley, Laurent Sauvage, Tarik Graba, and Yves Mathieu. Implementation and Evaluation of WDDL Countermeasures in FPGAs. In *CryptArchi*, Trégastel, France, June 1-4 2008. Trégastel, France; (abstract). 58

[104] Jean-Luc Danger, Olivier Meynard, Sylvain Guilley, Yu-Ichi Hayashi, and Naofumi Homma. *"Electromagnetic Radiation", chapter "Characterisation of the Information Leakage of Cryptographic Devices by using EM Analysis"*. InTech, 2012. ISBN: 978-953-51-0639-5. Available from: http://www.intechopen.com/books/electromagnetic-radiation/characterization-of-the-information-leakage-of-cryptographic-devices-by-using-em-analysis. 58

[105] Nicolas Darbel and Sylvain Guilley. Digital Matched Filter. United States Patent 7194021 issued in March 20, 2007; European Patent EP1355421; Patent application 01-LJ-118 (STMicroelectronics). 58

[106] Rémy Daudigny, Hervé Ledig, Frédéric Muller, and Frédéric Valette. SCARE of the DES. In *ACNS*, volume 3531 of *LNCS*, pages 393–406. Springer, June 2005. New York, NY, USA. 250, 252

[107] Nicolas Debande, Youssef Souissi, Aziz Elaabid, Sylvain Guilley, and Jean-Luc Danger. A Multiresolution Time-Frequency Analysis Based Side Channel Attacks (Poster). In *WIFS, IEEE Intl. Workshop on Information Forensics and Security*, November 29th - December 2nd 2011. Foz do Iguaçu, Brazil. 58

[108] Nicolas Debande, Youssef Souissi, Moulay Abdelaziz Elaabid, Sylvain Guilley, and Jean-Luc Danger. Wavelet Transform Based Pre-processing for Side Channel Analysis. In *HASP*, pages 32–38. IEEE, December 2nd 2012. Vancouver, British Columbia, Canada. DOI: 10.1109/MICROW.2012.15. 58

[109] Nicolas Debande, Youssef Souissi, Maxime Nassar, Sylvain Guilley, Thanh-Ha Le, and Jean-Luc Danger. "re-synchronization by moments": An efficient solution to align side-channel traces. In *WIFS*, pages 1–6. IEEE, 2011. 58

[110] Nicolas Debande, Youssef Souissi, Maxime Nassar, Thanh ha Le, Sylvain Guilley, and Jean-Luc Danger. Side Channel Analysis enhancement: A proposition for measurements resynchronisation. In *CryptArchi*, Bochum, Germany, June 15–18 2011. Bochum, Germany; (abstract). 58

[111] Amine Dehbaoui, Victor Lomne, Philippe Maurine, and Lionel Torres. Magnitude squared incoherence EM analysis for integrated cryptographic module localisation. *Electronics Letters*, 45(15):778–780, 16 2009. 4

[112] Whitfield Diffie and Martin Edward Hellman. New Directions in Cryptography. *IEEE Trans. on Info. Theory*, 22(6):644–654, 1976. 335

[113] Itai Dinur and Adi Shamir. Side Channel Cube Attacks on Block Ciphers. Cryptology ePrint Archive, Report 2009/127, March 2009. http://eprint.iacr.org/. 250

[114] Guillaume Duc, Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Tarik Graba, Yves Mathieu, and Renaud Pacalet. DPA contests. In *COSADE*, May 3rd 2012. Darmstadt, Germany. 57, 58

[115] Guillaume Duc, Sylvain Guilley, Laurent Sauvage, Florent Flament, Maxime Nassar, Nidhal Selmane, Jean-Luc Danger, Tarik Graba, Yves Mathieu, and Renaud Pacalet. Results of the 2009–2010 "DPA contest v2". In *COSADE*, February 2011. Darmstadt, Germany. (slides). 58

[116] Loïc Duflot, Philippe Le Moigne, and Fabien Germain. Device Forming a Logic Gate for Minimizing the Differences in Electrical or Electromagnetic Behavior in an Integrated Circuit Manipulating a Secret, November 9 2006. Patent from the État Francais, représenté par le secrétariat général de la défense nationale, WO/2006/117391, http://www.wipo.int/pctdb/en/wo.jsp?WO=2006117391. 86

[117] Marcia B. Costa e Silva, Qing Xu, Sébastien Agnolini, Sylvain Guilley, Jean-Luc Danger, Philippe Gallion, and Francisco J. Mendieta. Integrating a QPSK Quantum Key Distribution Link. In *ECOC*, September 24–28 2006. Cannes, France, DOI: 10.1109/ECOC.2006.4801094, http://arxiv.org/abs/quant-ph/0611102. 58

[118] Moulay Abdelaziz Elaabid and Sylvain Guilley. Practical Improvements of Profiled Side-Channel Attacks on a Hardware Crypto-Accelerator. In *AFRICACRYPT*, volume 6055 of *LNCS*, pages 243–260. Springer, May 03-06 2010. Stellenbosch, South Africa. DOI: 10.1007/978-3-642-12678-9_15. 12, 58, 235, 236, 238, 239, 241

[119] Moulay Abdelaziz Elaabid and Sylvain Guilley. Portability of Templates. *Journal of Cryptographic Engineering*, 2(1):63–74, 2012. DOI: 10.1007/s13389-012-0030-6. 58

[120] Moulay Aziz Elaabid, Sylvain Guilley, and Jean-Luc Danger. Exotic Leakage Models. In *CryptArchi*, Bochum, Germany, June 15–18 2011. Bochum, Germany; (abstract). 38, 58

[121] Paul N. Fahn and Peter K. Pearson. IPA: A New Class of Power Attacks. In *CHES*, volume 1717 of *LNCS*, page 173. Springer Berlin / Heidelberg, August 1999. Worcester, MA, USA. ISSN 0302-9743. 120

[122] Olivier Faurax, Assia Tria, Laurent Freund, and Frédéric Bancel. Robustness of circuits under delay-induced faults: test of AES with the PAFI tool. In *IOLTS*, pages 185–186. IEEE Computer Society, 8-11 July 2007. Heraklion, Crete, Greece. 201

[123] Olivier Faurax, Assia Tria, Laurent Freund, and Frédéric Bancel. Robustness of circuits under delay-induced faults: test of AES with the PAFI tool. *IEEE International On-Line Testing Symposium*, pages 185–186, July 8-11 2007. Heraklion, Crete, Greece. 271

[124] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy N. Rothblum. Leakage-Resilient Signatures. In *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 343–360. Springer, February 9-11 2010. Zurich, Switzerland. 331

[125] Laurent Fesquet, Jérôme Quartana, and Marc Renaudin. Asynchronous Systems on Programmable Logic. In *ReCoSoC*, pages 105–112, 2005. 104

[126] Florent Flament, Sumanta Chaudhuri, and Sylvain Guilley. La Loi de Rent et ses Applications au Placement/Routage. In *JNRDM*, volume 10, May 14–16 2007. Lille, France. ISSN 1774-0290, (Online PDF version). 58

[127] Florent Flament, Houssem Maghrebi, Moulay Aziz Elaabid, Jean-Luc Danger, Sylvain Guilley, and Laurent Sauvage. About Probability Density Function Estimation for Side Channel Analysis. In *COSADE*, pages 15–23, February 4-5 2010. Darmstadt, Germany. http://cosade2010.cased.de/files/proceedings/cosade2010_paper_4.pdf. 58

[128] Mike Fournigault, Pierre-Yvan Liardet, Yannick Teglia, Alain Trémeau, and Frédérique Robert-Inacio. Reverse Engineering of Embedded Software Using Syntactic Pattern Recognition. In *On the Move to Meaningful Internet Systems: OTM 2006 Workshops*, volume 4277 of *LNCS*, pages 527–536. Springer, 2006. Montpellier, France, DOI: 10.1007/11915034. 250, 252

[129] Julien Francq and Olivier Faurax. Security of several AES Implementations against Delay Faults. In *Proceedings of the 12th Nordic Workshop on Secure IT Systems (NordSec 2007)*, October 2007. Reykjavík, Iceland. 280

[130] Toshinori Fukunaga and Junko Takahashi. Practical fault attack on a cryptographic LSI with ISO/IEC 18033-3 block ciphers. In *FDTC*, pages 84–92. IEEE Computer Society, September 6th 2009. Lausanne, Switzerland. DOI: 10.1109/FDTC.2009.34. 306

[131] Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine Masking against Higher-Order Side Channel Analysis. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *LNCS*, pages 262–280. Springer, 2010. 12, 80

[132] 3rd Generation Smart Card Project, G3Card; European project under grant IST-1999-13515. Website: http://www.g3card.org/. 323

[133] Berndt M. Gammel and Stefan Mangard. On the duality of probing and fault attacks. Cryptology ePrint Archive, Report 2009/352, 2009. http://eprint.iacr.org/. 234, 306

[134] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic Analysis: Concrete Results. In *CHES*, volume 2162 of *LNCS*, pages 251–261. Springer, May 14-16 2001. Paris, France. 75, 105, 120, 162, 164, 206

[135] Flavio D. Garcia, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Wirelessly Pickpocketing a Mifare Classic Card. In *IEEE Symposium on Security and Privacy — S&P '09*, Oakland, California, USA, May 2009. IEEE. 253

[136] Catherine H. Gebotys. *Security in Embedded Devices*. Springer, 2010. ISBN: 978-1-4419-1529-0; DOI: 10.1007/978-1-4419-1530-6. 2

[137] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security against Hardware Tampering. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 258–277. Springer, February 19-21 2004. Cambridge, MA, USA. 311, 312

[138] Matteo Giaconia, Marco Macchetti, Francesco Regazzoni, and Kai Schramm. Area and Power Efficient Synthesis of DPA-Resistant Cryptographic S-Boxes. In *VLSI Design*, pages 731–737. IEEE Computer Society, 6-10 January 2007. Bangalore, India. 158

[139] Benedikt Gierlichs. DPA-Resistance Without Routing Constraints? – A Cautionary Note About MDPL Security –. In *CHES*, volume 4727 of *LNCS*, pages 107–120. Springer, September 2007. Vienna, Austria. 146

[140] Benedikt Gierlichs, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede. Revisiting Higher-Order DPA Attacks: Multivariate Mutual Information Analysis. In *CT-RSA*, volume 5985 of *LNCS*, pages 221–234. Springer, March 1-5 2010. San Francisco, CA, USA. 235

[141] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *CHES, 10th International Workshop*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, August 10-13 2008. Washington, D.C., USA. 77, 200, 210, 234, 242, 258

[142] Benedikt Gierlichs, Elke De Mulder, Bart Preneel, and Ingrid Verbauwhede. Empirical comparison of side channel analysis distinguishers on DES in hardware. In IEEE, editor, *ECCTD. European Conference on Circuit Theory and Design*, pages 391–394, August 23-27 2009. Antalya, Turkey. 23, 234

[143] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. Stochastic Methods. In *CHES*, volume 4249 of *LNCS*, pages 15–29. Springer, October 10-13 2006. Yokohama, Japan. 242

[144] Christophe Giraud and Hugues Thiebeauld. A Survey on Fault Attacks. In Kluwer, editor, *Smart Card Research and Advanced Applications VI, IFIP 18th, World Computer Congress, TC8/WG8.8 & TC11/WG11.2 Sixth International Conference on Smart Card Research and Advanced Applications (CARDIS)*, pages 159–176, 22-27 August 2004. Toulouse, France. 120, 318

[145] Martin Goldack. *Side Channel Based Reverse Engineering for Microcontrollers*. Ruhr-Universität-Bochum, Germany, January 2008. http://www.crypto.ruhr-uni-bochum.de/en_theses.html. 250, 252

[146] Kevin Gotze. A survey of frequently identified vulnerabilities in commercial computing semiconductors. In *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, pages 122 –126, june 2011. 2

[147] Louis Goubin and Jacques Patarin. Des and differential power analysis. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems*, volume 1717 of *LNCS*, pages 158–172. Springer, 1999. 85

[148] Louis Goubin and Jacques Patarin. DES and Differential Power Analysis. The "Duplication" Method. In *CHES*, LNCS, pages 158–172. Springer, Aug 1999. Worcester, MA, USA. 12, 80

[149] Sudhakar Govindavajhala and Andrew W. Appel. Using Memory Errors to Attack a Virtual Machine. In *SP'03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 154–165, Washington, DC, USA, May 11-14 2003. IEEE Computer Society. Berkeley, CA, USA. 306

[150] Alfred Grill, Sylvain Guilley, Vishnubhai Patel, and Katherina Babich. Effects of precursor additives on the stability of plasma enhanced chemical vapor deposition `a-GeC(O):H` films. *Journal of Materials Research*, 17(2):367–375, Feb 2002. DOI: 10.1557/JMR.2002.0052. 58

[151] Jorge Guajardo and Bart Mennink. On Side-Channel Resistant Block Cipher Usage. In Mike Burmester, Gene Tsudik, Spyros S. Magliveras, and Ivana Ilic, editors, *ISC*, volume 6531 of *LNCS*, pages 254–268. Springer, 2010. 338

[152] Jorge Guajardo and Bart Mennink. Towards Side-Channel Resistant Block Cipher Usage or Can We Encrypt Without Side-Channel Countermeasures? Cryptology ePrint Archive, Report 2010/015, January 11 2010. http://eprint.iacr.org/2010/015. 313

[153] Sylvain Guilley. Attaques SPA, DPA et DEMA sur un co-processeur DES : liens entre attaques réussies et architectures de circuits cryptographiques. In *Crypto'Puces 2007*. 15 – 18 avril 2007, Île de Porquerolles, France. 58

[154] Sylvain Guilley. Implantation d'un bit quantique dans un circuit supraconducteur. Master's thesis, Internship during the "quantum physics" MSc of LKB, at CEA/DRECAM (Saclay), Juin 2002. 41 pages, http://comelec.enst.fr/~guilley/dea.pdf. 58

[155] Sylvain Guilley. Caractérisation du canal caché "consommation instantanée" des portes CMOS. In *JNRDM*, volume 7, May 4–6 2004. Marseille, France. ISSN 1774-0290, (Online PDF version). 58

[156] Sylvain Guilley. Evaluation de différentes structures en transistors des portes "C-Element", May 2004. http://hal.archives-ouvertes.fr/hal-00707987. 58

[157] Sylvain Guilley. CMOS Structures and CAD Methods for the Design of DPA-proof ASICs. In *International Conference on Cryptographic Architectures Embedded in Reconfigurable Devices – CryptArchi 2005*, June 8–11 2005. Le Bessat near Saint-Étienne, http://cryptarchi.univ-st-etienne.fr/workshop05/. 58

[158] Sylvain Guilley. Implémentation d'un multiplieur de Montgomery sécurisé et cascadable. In *JNRDM*, volume 8, May 10–12 2005. Paris, France. ISSN 1774-0290, (Online PDF version). 58

[159] Sylvain Guilley. Attaques sur les implémentations des algorithmes de chiffrement symétrique, December 14 2006. Paris 8 University (MAATICAH) Seminar "protection de l'information", Room A 148, Saint-Denis, France. 58

[160] Sylvain Guilley. Geometrical Counter-Measures against Side-Channel Attacks, October 31st 2006. UCL seminar, Belevitch room, Louvain-la-Neuve, Belgium. 58

[161] Sylvain Guilley. Architecture et CAO pour Crypto processeurs sécurisés, February 19 2007. Journée Thématique : Groupe "Logiciels Embarqués et Architectures Matérielles", Université Pierre et Marie Curie, campus Jussieu, salle B202 de la maison de la pédagogie, Paris. 58

[162] Sylvain Guilley. *Geometrical Counter-Measures against Side-Channel Attacks.* PhD thesis, ENST / CNRS LTCI, January 2007. 219 pages; Id: 2007 E 003; Online versions: http://pastel.paristech.org/2562/ or http://www.iacr.org/phds/?p=detail&entry=708. 58, 91

[163] Sylvain Guilley. Logiciel "Cascaded-MMM", January 19 2010.
Dépôt auprès de l'APP numéro : IDDN.FR.001.040016.000.S.P.2010.000.20000. 58

[164] Sylvain Guilley. Resilience and Formal Proof, December 8 2010. Salon CARTES, Villepinte, France. 58

[165] Sylvain Guilley. Évaluation de contre-mesures aux attaques physiques, November 4 2010. Salle de séminaire du LIRMM, https://www.lirmm.fr/gt-secnum/index.php/seminaire. 58

[166] Sylvain Guilley. Cryptographic protocols resilient to physical level attacks, September 21-23 2011. eSmart, Sophia Antipolis, France. http://smart-event.eu/11/s-smart/program.htm. 58, 72

[167] Sylvain Guilley. Embedded Systems Attacks and Counter-Measures Strategies. In *JFFoE (Japan French Frontiers of Engineering). Security in ICT, session "Next Generation, Low power, Systems/Smart networks"*, February 25–28 2012. Kyōto, Japan. i, 58

[168] Sylvain Guilley. Resilience: A New Security Paradigm for Secure Elements, March 28–29 2012. CARTES in Asia Conference, AsiaWorld Expo in Hong Kong. 58

[169] Sylvain Guilley, Claude Carlet, Houssem Maghrebi, Jean-Luc Danger, and Emmanuel Prouff. Leakage Squeezing — Defeating Instantaneous $(d+1)$th-order Correlation Power Analysis with Strictly Less Than d Masks. In *CryptArchi*, Château de Goutelas, Marcoux, France, June 19–22 2012. Château de Goutelas, Marcoux, France; (abstract). 58

[170] Sylvain Guilley, Sumanta Chaudhuri, Philippe Hoogvorst, and Jean-Luc Danger. Balanced embedded FPGA architecture enabling efficient HW and SW counter-measures against physical attacks. July 2007. PASR USEIT'07, CNES, Toulouse, France. 58

[171] Sylvain Guilley, Sumanta Chaudhuri, Laurent Sauvage, Jean-Luc Danger, Taha Beyrouthy, and Laurent Fesquet. Updates on the Potential of Clock-Less Logics to Strengthen Cryptographic Circuits against Side-Channel Attacks. In *ICECS*, IEEE, pages 351–354, December 13–16 2009. Medina, Yasmine Hammamet, Tunisia. DOI: 10.1109/ICECS.2009.5411008. 58, 194, 323

[172] Sylvain Guilley, Sumanta Chaudhuri, Laurent Sauvage, Tarik Graba, Jean-Luc Danger, Philippe Hoogvorst, Ving-Nga Vong, Maxime Nassar, and Florent Flament. Shall we trust WDDL? In Vieweg+Teubner, editor, *Future of Trust in Computing*, volume 2, pages 208–215, Berlin, Germany, jun 2008. DOI: 10.1007/978-3-8348-9324-6_22. Berlin, Germany. 32, 58

[173] Sylvain Guilley, Sumanta Chaudhuri, Laurent Sauvage, Tarik Graba, Jean-Luc Danger, Philippe Hoogvorst, Vinh-Nga Vong, and Maxime Nassar. Place-and-Route Impact on the Security of DPL Designs in FPGAs. In *HOST (Hardware Oriented Security and Trust), IEEE*, pages 29–35, Anaheim, CA, USA, jun 2008. 58

[174] Sylvain Guilley, Sumanta Chaudhuri, Laurent Sauvage, Tarik Graba, Jean-Luc Danger, Philippe Hoogvorst, Vinh-Nga Vong, and Maxime Nassar. Place-and-Route Impact on the Security of DPL Designs in FPGAs. In *HOST*, pages 29–35. IEEE Computer Society, June 9 2008. Anaheim, USA. ISBN = 978-1-4244-2401-6. 187, 319

[175] Sylvain Guilley, Sumanta Chaudhuri, Laurent Sauvage, Philippe Hoogvorst, Renaud Pacalet, and Guido Marco Bertoni. Security Evaluation of WDDL and SecLib Countermeasures against Power Attacks. *IEEE Transactions on Computers*, 57(11):1482–1497, nov 2008. v, 40, 58, 119, 162, 164, 270, 323

[176] Sylvain Guilley and Jean-Luc Danger. Global Faults on Cryptographic Circuits. Chapter 17 of [233]. 58

[177] Sylvain Guilley and Jean-Luc Danger. Protection des modules de cryptographie contre les attaques sur les canaux cachés par chiffrement Vernam des fuites d'information, 20 Janvier 2009. Brevet Français FR09/50342, assigné à l'Institut TELECOM. 58

[178] Sylvain Guilley and Jean-Luc Danger. Technique de masquage personnalisé résistante aux attaques séquentielles d'ordre quelconque basées sur un changement de représentation linéaire moins coûteux que l'état-de-l'art., 2009. Brevet Français FR 09/58030. 19, 58, 208

[179] Sylvain Guilley and Jean-Luc Danger. Procédé de calcul cryptographique résilient aux attaques par injection de fautes, produit programme d'ordinateur et composant électronique correspondant, 28 Décembre 2011. Brevet Français. 41, 43, 58, 72

[180] Sylvain Guilley, Jean-Luc Danger, Moulay Abdelaziz El Aabid, Renaud Pacalet, and Philippe Hoogvorst. Symbolic Simulation for Security. July 2007. PASR USEIT'07, CNES, Toulouse, France. 58

[181] Sylvain Guilley, Jean-Luc Danger, Philippe Nguyen, Sébastien Briais, and Thibault Porteboeuf. Composant électronique comprenant un module de filtrage et de partitionnement, 28 Décembre 2011. Brevet Français. 43, 58

[182] Sylvain Guilley, Jean-Luc Danger, Robert Nguyen, and Philippe Nguyen. System-Level Methods to Prevent Reverse-Engineering, Cloning, and Trojan Insertion. In Sumeet Dua, Aryya Gangopadhyay, Parimala Thulasiraman, Umberto Straccia, Michael A. Shepherd, and Benno Stein, editors, *ICISTM (PPREW workshop)*, volume 285 of *Communications in Computer and Information Science*, pages 433–438. Springer, 2012. 58

[183] Sylvain Guilley, Jean-Luc Danger, and Laurent Sauvage. Protection du mécanisme de déchiffrement des fichiers de configuration pour FPGAs, 12 Août 2008. Brevet Français FR08/55536, assigné à l'Institut TELECOM; WO/2010/018072. 58, 293

[184] Sylvain Guilley and Anh Duc Dao. Logiciel "`genlut`", 18 apr 2008. Dépôt auprès de l'APP numéro : IDDN.FR.001.160027.000.S.P.2008.000.20600. 58

[185] Sylvain Guilley and Anh Duc Dao. Logiciel "`vDuplicate`", 18 apr 2008. Dépôt auprès de l'APP numéro : IDDN.FR.001.160028.000.S.P.2008.000.20600. 58, 273

[186] Sylvain Guilley, Florent Flament, Yves Mathieu, and Renaud Pacalet. Security Evaluation of a Balanced Quasi-Delay Insensitive Library. In *DCIS*, Grenoble, France, nov 2008. IEEE. Session 5D – Reliable and Secure Architectures, ISBN: 978-2-84813-124-5. Available on-line: `http://hal.archives-ouvertes.fr/hal-00283405/en/`. v, 58, 85

[187] Sylvain Guilley, Florent Flament, Renaud Pacalet, Philippe Hoogvorst, and Yves Mathieu. Secured CAD Back-End Flow for Power-Analysis Resistant Cryptoprocessors. *IEEE Design & Test of Computers, special issue on "Design and Test of ICs for Secure Embedded Computing"*, 24(6):546–555, November-December 2007. DOI: 10.1109/MDT.2007.202. v, 58, 103

[188] Sylvain Guilley, Florent Flament, Renaud Pacalet, Philippe Hoogvorst, and Yves Mathieu. Secured CAD Back-End Flow for Power-Analysis Resistant Cryptoprocessors. *IEEE Design & Test of Computers, special issue on "Design and Test of ICs for Secure Embedded Computing"*, 24(6):546–555, November-December 2007. 86, 90, 124, 125, 132, 162, 164

[189] Sylvain Guilley, Florent Flament, Renaud Pacalet, Philippe Hoogvorst, and Yves Mathieu. Security Evaluation of a Balanced Quasi-Delay Insensitive Library. In *DCIS*, Grenoble, France, nov 2008. IEEE. 6 pages, Session 5D – Reliable and Secure Architectures, ISBN: 978-2-84813-124-5, full text in HAL: http://hal.archives-ouvertes.fr/hal-00283405/en/. 125, 128, 157, 158, 200, 270, 295, 323

[190] Sylvain Guilley and Philippe Hoogvorst. The Proof by $2^M - 1$: a Low-Cost Method to Check Arithmetic Computations. In *IFIP Advances in Information and Communication Technology, SEC*, volume IFIP 181/2005, pages 589–600, Makuhari-Messe, Chiba, Japan, may 2005. Makuhari-Messe, Chiba, Japan. DOI: 10.1007/0-387-25660-1_39. 58

[191] Sylvain Guilley, Philippe Hoogvorst, Yves Mathieu, and Renaud Pacalet. The "Backend Duplication" Method. In *CHES*, volume 3659 of *LNCS*, pages 383–397. Springer, 2005. August 29th – September 1st, Edinburgh, Scotland, UK. 33, 34, 58, 109, 122, 131, 160, 162, 174, 294, 319

[192] Sylvain Guilley, Philippe Hoogvorst, Yves Mathieu, and Renaud Pacalet. The "Backend Duplication" Method. In LNCS, editor, *CHES*, volume 3659, pages 383–397, August 2005. Edinburgh, Scotland, UK. 90

[193] Sylvain Guilley, Philippe Hoogvorst, Yves Mathieu, Renaud Pacalet, and Jean Provost. CMOS Structures Suitable for Secured Hardware. In *DATE'04 – Volume 2*, pages 1414–1415. IEEE Computer Society, February 2004. Paris, France. DOI: 10.1109/DATE.2004.1269113 (Online version). 58, 86, 104, 107, 108, 121, 125, 158, 295, 323

[194] Sylvain Guilley, Philippe Hoogvorst, and Renaud Pacalet. Differential Power Analysis Model and some Results. In Kluwer, editor, *Proceedings of WCC/CARDIS*, pages 127–142, Aug 2004. Toulouse, France. DOI: 10.1007/1-4020-8147-2_9. 21, 58, 165

[195] Sylvain Guilley, Philippe Hoogvorst, and Renaud Pacalet. A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation. *Integration, The VLSI Journal*, 40(4):479–489, July 2007. DOI: 10.1016/j.vlsi.2006.06.004. 24, 58, 111, 116, 124, 149, 194

[196] Sylvain Guilley, Philippe Hoogvorst, Renaud Pacalet, and Johannes Schmidt. Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties. In Presse Universitaire de Rouen et du Havre, editor, *BFCA*, pages 1–25, 2007. May 02–04, Paris, France, http://www.liafa.jussieu.fr/bfca/books/BFCA07.pdf. 58, 139, 165, 198, 262

[197] Sylvain Guilley, Karim Khalfallah, Victor Lomne, and Jean-Luc Danger. Formal Framework for the Evaluation of Waveform Resynchronization Algorithms. In LNCS, editor, *WISTP: Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing*, volume 6633 of *LNCS*, pages 100–115. Springer, June 1-3 2011. Heraklion, Greece. DOI: 10.1007/978-3-642-21040-2_7. 58

[198] Sylvain Guilley, Houssem Maghrebi, Youssef Souissi, Laurent Sauvage, and Jean-Luc Danger. Quantifying the Quality of Side-Channel Acquisitions. In *COSADE*, pages 16–28, February 24-25 2011. Darmstadt, Germany. http://cosade2011.cased.de/files/2011/cosade2011_talk2_paper.pdf. 58

[199] Sylvain Guilley, Olivier Meynard, Maxime Nassar, Guillaume Duc, Philippe Hoogvorst, Houssem Maghrebi, Aziz Elaabid, Shivam Bhasin, Youssef Souissi, Nicolas Debande, Laurent Sauvage, and Jean-Luc Danger. Vade Mecum on Side-Channels Attacks and Countermeasures for the Designer and the Evaluator. In *DTIS (Design & Technologies of Integrated Systems)*, IEEE. IEEE, March 6-8 2011. Athens, Greece. DOI: 10.1109/DTIS.2011.5941419 ; Online version: http://hal.archives-ouvertes.fr/hal-00579020/en/. iv, 5, 9, 58, 73

[200] Sylvain Guilley, Olivier Meynard, Laurent Sauvage, and Jean-Luc Danger. An Empirical Study of the EIS Assumption in Side Channel Attacks against Hardware Implementations. In *COSADE*, pages 10–14, February 4-5 2010. Darmstadt, Germany. http://cosade2010.cased.de/files/proceedings/cosade2010_paper_3.pdf. 11, 12, 58

[201] Sylvain Guilley and Philippe Nguyen. Smart-SIC Analyzer: A Circuit-Level Vulnerability Assistant, September 21-24 2010. eSmart, Sophia Antipolis, France. http://smart-event.eu/10/s-smart/program.htm. 58

[202] Sylvain Guilley, Philippe Nguyen, Robert Nguyen, Hassan Triqui, and Jean-Luc Danger. Smart-SIC Analyzer, September 26-27 2011. Panel Discussion – Tool Vendor / Laboratory. Non-Invasive Attack Testing Workshop (NIAT 2011), co-organized by NIST & AIST. Todai-ji Cultural Center, Nara, Japan. (PDF). 58

[203] Sylvain Guilley and Renaud Pacalet. SoC Security: a War against Side-Channels. *Annals of the Telecommunications*, 59(7-8):998–1009, July-August 2004. ISSN 0003-4347. DOI: 10.1007/BF03180031. 3, 58

[204] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Tarik Graba, and Yves Mathieu. Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGAs. In *SSIRI*, pages 16–23, Yokohama, Japan, jul 2008. IEEE Computer Society. DOI: 10.1109/SSIRI.2008.31, http://hal.archives-ouvertes.fr/hal-00259153/en/. 33, 58, 80, 272, 294

[205] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, and Philippe Hoogvorst. Area Optimization of Cryptographic Co-Processors Implemented in Dual-Rail with Precharge Positive Logic. In *FPL (18th IEEE International Conference on Field-Programmable Logic and Applications)*, pages 161–166, Heidelberg, Germany, sep 2008. ISBN: 978-1-4244-1961-6. 35, 58, 272

[206] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, and Nidhal Selmane. Fault Injection Resilience. In *FDTC*, pages 51–65. IEEE Computer Society, August 21 2010. Santa Barbara, CA, USA. DOI: 10.1109/FDTC.2010.15; Complete version: http://hal.archives-ouvertes.fr/hal-00482194/en/. viii, 9, 43, 58, 84, 305

[207] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, and Nidhal Selmane. Fault Injection Resilience. In *FDTC*, pages 51–65. IEEE, August 21 2010. Santa Barbara, CA, USA. 334

[208] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Nidhal Selmane, and Renaud Pacalet. Silicon-level solutions to counteract passive and active attacks. In *FDTC, 5th Workshop on Fault Detection and Tolerance in Cryptography, IEEE-CS*, pages 3–17, Washington DC, USA, aug 2008. 36, 58, 133, 199

[209] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Nidhal Selmane, and Denis Réal. Performance Evaluation of Protocols Resilient to Physical Attacks. In *HOST*, IEEE Computer Society, pages 51–56, June 5-6 2011. Convention Center, San Diego, California, USA. DOI: 10.1109/HST.2011.5954995. ix, 41, 43, 58, 329

[210] Sylvain Guilley, Laurent Sauvage, Florent Flament, Philippe Hoogvorst, and Renaud Pacalet. Evaluation of Power-Constant Dual-Rail Logics Counter-Measures against DPA with Design-Time Security Metrics. *IEEE Transactions on Computers*, 9(59):1250–1263, September 2010. DOI: 10.1109/TC.2010.104. vi, 35, 36, 37, 40, 58, 155, 323

[211] Sylvain Guilley, Laurent Sauvage, Florent Flament, Maxime Nassar, Nidhal Selmane, Jean-Luc Danger, Tarik Graba, Yves Mathieu, and enaud Pacalet. Mid-Term Report of the DPA Contest. In *CryptArchi*, Prague, Czech Republic, June 24th–27th 2009. Prague, Czech Republic; (abstract). 58

[212] Sylvain Guilley, Laurent Sauvage, Florent Flament, Maxime Nassar, Nidhal Selmane, Jean-Luc Danger, Tarik Graba, Yves Mathiew, and Renaud Pacalet. Overview of the 2008-2009 'DPA contest', September 6-9 2009. CHES Special Session 1: DPA Contest. Lausanne, Switzerland, (slides). 58

[213] Sylvain Guilley, Laurent Sauvage, Florent Flament, Maxime Nassar, Nidhal Selmane, Jean-Luc Danger, Philippe Hoogvorst, Tarik Graba, Yves Mathieu, and Renaud Pacalet. FPGAs for Counter-Measures Evaluation. In *PASTIS (PAca Security Trends In embedded Security)*, Gardanne (École des Mines de Saint-Étienne), France, dec 2nd 2008. http://www.secure-ic.com/PDF/pastis08_slides.pdf. 58

[214] Sylvain Guilley, Laurent Sauvage, Florent Flament, Maxime Nassar, Nidhal Selmane, Jean-Luc Danger, Philippe Hoogvorst, Tarik Graba, Yves Mathieu, and Renaud Pacalet. On the Power of Power Analyses. Invited talk at the ALI (ENSTA) and SALSA (LIP6/INRIA) seminar, March 6 2009. LIP6, room 847, http://uma.ensta-paristech.fr/conf/ali-salsa/slides/slides_sylvain_guilley.pdf. 58

[215] Sylvain Guilley, Laurent Sauvage, Julien Micolod, Denis Réal, and Frédéric Valette. Defeating Any Secret Cryptography with SCARE Attacks. In *LatinCrypt*, volume 6212 of *LNCS*, pages 273–293. Springer, August 8-11 2010. Puebla, México, DOI: 10.1007/978-3-642-14712-8_17. vii, 58, 249

[216] Helena Handschuh, Pascal Paillier, and Jacques Stern. Probing Attacks on Tamper-Resistant Devices. In *CHES*, volume 1717 of *LNCS*, pages 303–315. Springer, August 12-13 1999. Worcester, MA, USA. 9, 306

[217] Neil Hanley, Robert McEvoy, Michael Tunstall, Claire Whelan, Colin Murphy, and William P. Marnane. Correlation Power Analysis of Large Word Sizes. In *ISSC (Irish Signals and System Conference)*, pages 145–150. IET, 13-14 Sept 2007. Edinburgh, Scotland, UK. 186

[218] Wei He, Eduardo De La Torre, and Teresa Riesgo. A Precharge-Absorbed DPL Logic for Reducing Early Propagation Effects on FPGA Implementations. In *ReConFig*, pages 217–222. IEEE Computer Society, November 30 – December 2 2011. Cancún, Quintana Roo, México. DOI: 10.1109/ReConFig.2011.3. 33

[219] Philippe Hoogvorst. The Variance Power Attack. In *COSADE*, pages 4–9, February 4-5 2010. Darmstadt, Germany. http://cosade2010.cased.de/files/proceedings/cosade2010_paper_2.pdf. 77

[220] Philippe Hoogvorst, Jean-Luc Danger, and Guillaume Duc. Software Implementation of Dual-Rail Representation. In *COSADE*, February 24-25 2011. Darmstadt, Germany. 34, 80

[221] Philippe Hoogvorst, Sylvain Guilley, Sumanta Chaudhuri, Jean-Luc Danger, Taha Beyrouthy, and Laurent Fesquet. A Reconfigurable Programmable Logic Block for a Multi-Style Asynchronous FPGA resistant to Side-Channel Attacks. *CoRR*, abs/0809.3942, September 2008. 58

[222] Philippe Hoogvorst, Sylvain Guilley, Sumanta Chaudhuri, Jean-Luc Danger, Alin Razafindraibe, Taha Beyrouthy, Laurent Fesquet, and Marc Renaudin. A Reconfigurable Cell for a Multi-Style Asynchronous FPGA. pages 15–22, June 2007. ReCoSoC, Montpellier, France. http://arxiv.org/abs/0809.3942. 58, 104

[223] Philippe Hoogvorst, Sylvain Guilley, and Tarik Graba. Logiciel "`fpgasbox`", 09 july 2008. Dépôt auprès de l'APP numéro : IDDN.FR.001.280019.000.S.P.2008.000.20600. 58

[224] TELECOM ParisTech & Secure-IC. EveSoC, a side-channel eavesdropping system-on-chip, http://sourceforge.net/projects/evesoc/ (available from SourceForge under GNU Public License), 2009. 187, 302

[225] David Hwang, Kris Tiri, Alireza Hodjat, Bo-Cheng Lai, Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede. AES-Based Security Coprocessor IC in 0.18-μm CMOS With Resistance to Differential Power Analysis Side-Channel Attacks. *IEEE Journal of Solid-State Circuits*, 41(4):781–792, April 2006. Digital Object Identifier: 10.1109/JSSC.2006.870913. 114, 116

[226] IEEE. Delay and power calculation standards - Part 3: Standard Delay Format (SDF) for the electronic design process. *IEC 61523-3 First edition 2004-09; IEEE 1497*, pages 1–94, 2004. 32

[227] Makoto Ikeda, Hiroshi Yamauchi, and Kunihiro Asada. Tamper Resistivity Analysis for Nanometer LSI with Process Variations. In *ICECS*, pages 387–390, 2006. 93

[228] Institute of Electrical and Electronics Engineers (http://www.ieee.org/). IEEE Standard Verilog Description Language, Std 1364-2001, September 28 2001. ISBN: 0-7381-2826-0. 7

[229] Institute of Electrical and Electronics Engineers (http://www.ieee.org/). IEEE Standard VHDL (Very High Speed Integrated Circuits Description Language) Reference Manual, May 17 2002. ISBN: 0-7381-3247-0. 7, 174, 283

[230] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private Circuits II: Keeping Secrets in Tamperable Circuits. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 308–327. Springer, May 28 – June 1 2006. St. Petersburg, Russia. 320

[231] Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics, 2002. ISBN: 0387954422. 139

[232] Marc Joye, Pascal Paillier, and Berry Schoenmakers. On Second-Order Differential Power Analysis. In *CHES*, volume 3659 of *LNCS*, pages 293–308. Springer, August 29 – September 1st 2005. Edinburgh, UK. 246

[233] Marc Joye and Michael Tunstall. *Fault Analysis in Cryptography*. Springer LNCS, March 2011. http://joye.site88.net/FAbook.html. DOI: 10.1007/978-3-642-29656-7 ; ISBN 978-3-642-29655-0. 3, 289, 309, 354

[234] Marc Joye and Sung-Ming Yen. The Montgomery Powering Ladder. In Burton S. Kaliski, Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 291–302. Springer, 2002. 79

[235] Mohaned Kafi, Sylvain Guilley, Sandra Marcello, and David Naccache. Deconvolving Protected Signals. In *ARES/CISIS*, pages 687–694, Fukuoka, Kyūshū, Japan, March, 16th – 19th 2009. IEEE Computer Society Press. DOI: 10.1109/ARES.2009.197. 58

[236] Mark Karpovsky, Konrad J. Kulikowski, and Alexander Taubin. Robust Protection against Fault-Injection Attacks on Smart Cards Implementing the Advanced Encryption Standard. *IEEE Transactions on Computer-Aided Design*, 21(2), may 2004. 270

[237] Mark G. Karpovsky, Konrad J. Kulikowski, and Alexander Taubin. Robust Protection against Fault Injection Attacks on Smart Cards Implementing the Advanced Encryption Standard. In *DSN*, pages 93–101. IEEE Computer Society, June 28 – July 01 2004. Florence, Italy. 323, 324

[238] Peter Karsmakers, Benedikt Gierlichs, Kristiaan Pelckmans, Katrien De Cock, Johan Suykens, Bart Preneel, and Bart De Moor. Side channel attacks on cryptographic devices as a classification problem. COSIC technical report. 28

[239] Markus Kasper, Timo Kasper, Amir Moradi, and Christof Paar. Breaking KeeLoq in a Flash: On Extracting Keys at Lightning Speed. In Bart Preneel, editor, *AFRICACRYPT*, volume 5580 of *LNCS*, pages 403–420. Springer, 2009. 76

[240] Auguste Kerckhoffs. La cryptographie militaire (1). *Journal des sciences militaires*, 9:5–38, January 1883. http://en.wikipedia.org/wiki/Kerckhoffs_law. 1, 250

[241] Auguste Kerckhoffs. La cryptographie militaire (2). *Journal des sciences militaires*, 9:161–191, February 1883. http://en.wikipedia.org/wiki/Kerckhoffs_law. 1

[242] Farouk Khelil, Mohamed Hamdi, Sylvain Guilley, Jean-Luc Danger, and Nidhal Selmane. Fault Analysis Attack on an FPGA AES Implementation. In *NTMS*, pages 1–5, Tangier, Morocco, nov 2008. IEEE. DOI: 10.1109/NTMS.2008.ECP.45. 58, 201, 270, 280

[243] Lars R. Knudsen and Matthew Robshaw. *The Block Cipher Companion*. Information security and cryptography. Springer, 2011. 3

[244] Kaya Çetin Koç. *Cryptographic Engineering*. Springer US, 2009. 193

[245] Paul C. Kocher. Leak-resistant cryptographic indexed key update, March 25 2003. United States Patent 6,539,092 filed on July 2nd, 1999 at San Francisco, CA, USA. 41, 308, 332, 338

[246] Paul C. Kocher. Design and Validation Strategies for Obtaining Assurance in Countermeasures to Power Analysis and Related Attacks, September 26-29 2005. Honolulu, Hawai, USA; NIST's Physical Security Testing Workshop. Website: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-3/physec/physecdoc.html. 331, 340

[247] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Proceedings of CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer-Verlag, 1996. (PDF). 74, 76, 79

[248] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Proceedings of CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999. 36, 37, 75, 77, 85, 156, 164, 198, 234, 270, 292, 317

[249] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *CRYPTO*, volume 1666 of *LNCS*, pages pp 388–397. Springer, 1999. 120, 192

[250] Paul C. Kocher, Joshua M. Jaffe, and Benjamin C. Jun. Differential power analysis method and apparatus, September 8 2009. United States Patent, number 7,587,044. 2

[251] Oliver Kömmerling and Markus G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *WOST'99: Proceedings of the USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology*, pages 2–2, Berkeley, CA, USA, 1999. USENIX Association. (On-line paper). 313

[252] Boris Köpf and David Basin. An information-theoretic model for adaptive side-channel attacks. In *CCS'07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 286–296, New York, NY, USA, 2007. ACM. 209

[253] Konrad J. Kulikowski, Mark G. Karpovsky, and Alexander Taubin. Power Attacks on Secure Hardware Based on Early Propagation of Data. In *IOLTS*, pages 131–138. IEEE Computer Society, 2006. Como, Italy. 294, 319

[254] Ian Kuon and Jonathan Rose. Measuring the Gap Between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(2):203–215, February 2007. 256

[255] Thanh-Ha Le. *Analyses et Mesures Avancées du Rayonnement électromagnétique d'un Circuit Intégré*. PhD thesis, "Institut National Polytechnique" (INP), September 5 2007. Grenoble, France. 235

[256] Thanh-Ha Le and Maël Berthier. Mutual Information Analysis under the View of Higher-Order Statistics. In Isao Echizen, Noboru Kunihiro, and Ryôichi Sasaki, editors, *IWSEC*, volume 6434 of *LNCS*, pages 285–300. Springer, 2010. 28

[257] Thanh-Ha Le, Cécile Canovas, and Jessy Clédière. An overview of side channel analysis attacks. In *ASIACCS*, pages 33–43. ASIAN ACM Symposium on Information, Computer and Communications Security, 2008. DOI: 10.1145/1368310.1368319. Tōkyō, Japan. 5, 234

[258] Thanh-Ha Le, Jessy Clédière, Cécile Canovas, Bruno Robisson, Christine Servière, and Jean-Louis Lacoume. A Proposition for Correlation Power Analysis Enhancement. In *CHES*, volume 4249 of *LNCS*, pages 174–186. Springer, 2006. Yokohama, Japan. 120, 138, 259

[259] Thanh-Ha Le, Jessy Cledière, Christine Servière, and Jean-Louis Lacoume. How can Signal Processing Benefit Side Channel Attacks? In *Proceedings of IEEE Workshop on Signal Processing Applications for Public Security and Forensics (SAFE)*, pages 1–7, April 11-13 2007. Washington D.C., USA. 235

[260] Régis Leveugle. Early Analysis of Fault-based Attack Effects in Secure Circuits. *IEEE Trans. Computers*, 56(10):1431–1434, 2007. 323

[261] Huiyun Li. *Security evaluation at design time for cryptographic hardware*. PhD thesis, University of Cambridge, UK, April 2006. (Report UCAM-CL-TR-665, http://www.cl.cam.ac.uk/techreports/). 96

[262] Huiyun Li, A. Theodore Markettos, and Simon W. Moore. A security evaluation methodology for smart cards against electromagnetic analysis. In *Security Technology, 2005. CCST'05. 39th Annual 2005 International Carnahan Conference on*, pages 208–211, 11-14 Oct. 2005. 187

[263] Yang Li, Shigeto Gomisawa, Kazuo Sakiyama, and Kazuo Ohta. An Information Theoretic Perspective on the Differential Fault Analysis against AES. Cryptology ePrint Archive, Report 2010/032, 2010. http://eprint.iacr.org/. 308

[264] Yang Li, Kazuo Ohta, and Kazuo Sakiyama. Revisit fault sensitivity analysis on WDDL-AES. In *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, pages 148–153, june 2011. 40

[265] Yang Li, Kazuo Sakiyama, Lejla Batina, D. Nakatsu, and Kazuo Ohta. Power Variance Analysis breaks a masked ASIC implementation of AES. In *DATE*, pages 1059–1064. IEEE, March 8-12 2010. Dresden, Germany. 77

[266] Yang Li, Kazuo Sakiyama, Shigeto Gomisawa, Toshinori Fukunaga, Junko Takahashi, and Kazuo Ohta. Fault Sensitivity Analysis. In *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 320–334. Springer, August 17-20 2010. Santa Barbara, CA, USA. 39, 289, 316, 320

[267] Lang Lin and Wayne P. Burleson. Analysis and mitigation of process variation impacts on Power-Attack Tolerance. In *DAC*, pages 238–243. ACM, 2009. 37

[268] Hongying Liu, Guoyu Qian, Satoshi Goto, and Yukiyasu Tsunoo. Correlation Power Analysis Based on Switching Glitch Model. In Yongwha Chung and Moti Yung, editors, *WISA*, volume 6513 of *Lecture Notes in Computer Science*, pages 191–205. Springer, 2010. 32

[269] Lorentz Center, International Center for workshops in the Sciences. Workshop on "Provable Security against Physical Attacks", February 10-19 2010. Amsterdam, Netherlands. http://www.lorentzcenter.nl/lc/web/2010/383/program.php3?wsid=383. 308

[270] Yuanlin Lu and Vishwani D. Agrawal. CMOS Leakage and Glitch Minimization for Power-Performance Tradeoff. *Journal of Low Power Electronics*, 2(3):378–387, 2006. 32

[271] François Macé, François-Xavier Standaert, and Jean-Jacques Quisquater. Information theoretic evaluation of side-channel resistant logic styles. In *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 427–442. Springer, September 2007. Vienna, Austria. 183, 242

[272] François Macé, François-Xavier Standaert, Jean-Jacques Quisquater, and Jean-Didier Legat. A Design Methodology for Secured ICs Using Dynamic Current Mode Logic. In *PATMOS*, volume 3728 of *Lecture Notes in Computer Science*, pages 550–560. Springer, September 21–23 2005. Leuven, Belgium. 125

[273] Houssem Maghrebi, Claude Carlet, Sylvain Guilley, and Jean-Luc Danger. Optimal First-Order Masking with Linear and Non-linear Bijections. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *AFRICACRYPT*, volume 7374 of *Lecture Notes in Computer Science*, pages 360–377. Springer, 2012. 58

[274] Houssem Maghrebi, Jean-Luc Danger, Florent Flament, and Sylvain Guilley. Evaluation of Countermeasures Implementation Based on Boolean Masking to Thwart First and Second Order Side-Channel Attacks. In *SCS*, IEEE, pages 1–6, November 6–8 2009. Jerba, Tunisia. DOI: 10.1109/ICSCS.2009.5412597. 13, 30, 58, 77

[275] Houssem Maghrebi, Jean-Luc Danger, and Sylvain Guilley. Leakage Squeezing Countermeasure Against High Order Attacks. In *CryptArchi*, Gif-sur-Yvette, France, June 27-30 2010. Gif-sur-Yvette, France; (abstract). 58

[276] Houssem Maghrebi, Sylvain Guilley, Claude Carlet, and Jean-Luc Danger. Classification of High-Order Boolean Masking Schemes and Improvements of their Efficiency. Cryptology ePrint Archive, Report 2011/520, September 2011. http://eprint.iacr.org/2011/520. 16, 58

[277] Houssem Maghrebi, Sylvain Guilley, Claude Carlet, and Jean-Luc Danger. Optimal First-Order Masking with Linear and Non-Linear Bijections. In *AFRICACRYPT*, LNCS. Springer, July 10-12 2012. Al Akhawayn University in Ifrane, Morocco. 16

[278] Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger. Formal Security Evaluation of Hardware Boolean Masking against Second-Order Attacks. In *HOST*, IEEE Computer Society, pages 40–46, June 5-6 2011. Convention Center, San Diego, California, USA. DOI: 10.1109/HST.2011.5954993. 49, 58

[279] Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger. Leakage Squeezing Countermeasure Against High-Order Attacks. In *WISTP*, volume 6633 of *LNCS*, pages 208–223. Springer, June 1-3 2011. Heraklion, Greece. DOI: 10.1007/978-3-642-21040-2_14. 42

[280] Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger. Leakage Squeezing Countermeasure Against High-Order Attacks. In *WISTP*, volume 6633 of *LNCS*, pages 208–223. Springer, June 1-3 2011. (**best paper award**). Heraklion, Greece. DOI: 10.1007/978-3-642-21040-2_14. 58

[281] Houssem Maghrebi, Sylvain Guilley, Jean-Luc Danger, and Florent Flament. Entropy-based Power Attack. In *HOST*, IEEE Computer Society, pages 1–6, June 13-14 2010. Anaheim Convention Center, Anaheim, CA, USA. DOI: 10.1109/HST.2010.5513124. 30, 58, 77

[282] Houssem Maghrebi, Emmanuel Prouff, Sylvain Guilley, and Jean-Luc Danger. A First-Order Leak-Free Masking Countermeasure. In *CT-RSA*, volume 7178 of *LNCS*, pages 156–170. Springer, February 27 – March 2 2012. San Francisco, CA, USA. DOI: 10.1007/978-3-642-27954-6_10. 16, 18, 42, 58

[283] Houssem Maghrebi, Emmanuel Prouff, Sylvain Guilley, and Jean-Luc Danger. A First-Order Leak-Free Masking Countermeasure. Cryptology ePrint Archive, Report 2012/028, 2012. http://eprint.iacr.org/2012/028. 16

[284] Houssem Maghrebi, Emmanuel Prouff, Sylvain Guilley, and Jean-Luc Danger. Register Leakage Masking Using Gray Code. In *HOST*, IEEE Computer Society, pages 37–42, June 2-3 2012. Moscone Center, San Francisco, CA, USA. DOI: 10.1109/HST.2012.6224316. 58

[285] Houssem Maghrebi, Olivier Rioul, Sylvain Guilley, and Jean-Luc Danger. Comparison between Side Channel Analysis Distinguishers. In Tat Wing Chim and Tsz Hon Yuen, editors, *ICICS*, volume 7618 of *LNCS*, pages 331–340. Springer, October 29-31 2012. Hong Kong. 58, 77

[286] Vincent Maingot, Jean-Baptiste Ferron, Régis Leveugle, Vincent Pouget, and Alexandre Douin. Configuration errors analysis in SRAM-based FPGAs: software tool and practical results. *Microelectronics Reliability*, 47(9-11):1836–1840, 2007. 284

[287] Vincent Maingot and Régis Leveugle. Influence of error detecting or correcting codes on the sensitivity to DPA of an AES S-box. In *SCS*, IEEE, pages 1–5, November 6–8 2009. Jerba, Tunisia. DOI: 10.1109/ICSCS.2009.5412600. 323

[288] Paolo Maistri and Régis Leveugle. Double-data-rate computation as a countermeasure against fault analysis. *IEEE Trans. Comput.*, 57(11):1528–1539, 2008. 270, 288

[289] Tal Malkin, François-Xavier Standaert, and Moti Yung. A Comparative Cost/Security Analysis of Fault Attack Countermeasures. In *FDTC*, volume 4236 of *Lecture Notes in Computer Science*, pages 159–172. Springer, October 10 2006. Yokohama, Japan. 307, 323

[290] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, http://www.dpabook.org/. 2, 75, 78, 120, 206, 234, 318

[291] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for All - All for One: Unifying Standard DPA Attacks. Cryptology ePrint Archive, Report 2009/449, 2009. 28, 75

[292] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for All - All for One: Unifying Standard DPA Attacks. *Information Security, IET*, 5(2):100–111, 2011. ISSN: 1751-8709 ; Digital Object Identifier: 10.1049/iet-ifs.2010.0096. 28

[293] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In *CT-RSA*, volume 3376 of *LNCS*, pages 351–365. Springer, 2005. San Francisco, CA, USA. 85, 146

[294] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully Attacking Masked AES Hardware Implementations. In LNCS, editor, *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 157–171. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK. 146, 165, 272

[295] Stefan Mangard and Kai Schramm. Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In *CHES*, volume 4249 of *LNCS*, pages 76–90. Springer, October 10-13 2006. Yokohama, Japan. 32, 207

[296] A. Theodore Markettos and Simon W. Moore. The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators. In Christophe Clavier and Kris Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 317–331. Springer, 2009. 9

[297] H. Marzouqi, K. Salah, M. Al-Qutayri, and M. C. Y. Yeun. A Unified Countermeasure Against Side Channel Attacks on Cryptographic RFID. In *Internet Technology and Secured Transactions (ICITST)*, pages 13–18. IEEE, December 11-14 2011. Abu Dhabi, United Arab Emirates. ISBN: 978-1-4577-0884-8. 293

[298] Mitsuru Matsui. Linear Cryptoanalysis Method for DES Cipher. In Tor Helleseth, editor, *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993. 1

[299] T. Matsumoto, H. Mimura, and D. Suzuki. Complementary logics vs masked logics: Which countermeasure is a better selection? In IEEE, editor, *ECCTD. European Conference on Circuit Theory and Design*, pages 399–402, August 23-27 2009. Antalya, Turkey. 42

[300] Robert P. McEvoy, Colin C. Murphy, William P. Marnane, and Michael Tunstall. Isolated WDDL: A Hiding Countermeasure for Differential Power Analysis on FPGAs. *ACM Trans. Reconfigurable Technol. Syst.*, 2(1):1–23, 2009. 33

[301] Robert P. McEvoy, Colin C. Murphy, William P. Marnane, and Michael Tunstall. Isolated WDDL: A Hiding Countermeasure for Differential Power Analysis on FPGAs. *ACM Trans. Reconfigurable Technol. Syst. (TRETS)*, 2(1):1–23, 2009. 323, 324

[302] Robert P. McEvoy, Michael Tunstall, Claire Whelan, Colin C. Murphy, and William P. Marnane. A differential side-channel analysis countermeasure. European Patent Application (EP 2148462 A1), filled in 27.01.2010. 315, 335

[303] Robert P. McEvoy, Michael Tunstall, Claire Whelan, Colin C. Murphy, and William P. Marnane. All-or-Nothing Transforms as a Countermeasure to Differential Side-Channel Analysis. Cryptology ePrint Archive, Report 2009/185, April 30 2009. http://eprint.iacr.org/2009/185. 315, 335

[304] Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh Re-Keying: Security against Side-Channel and Fault Attacks for Low-Cost Devices. In *AFRICACRYPT*, volume 6055 of *LNCS*, pages 279–296. Springer, May 03-06 2010. Stellenbosch, South Africa. DOI: 10.1007/978-3-642-12678-9_17. 41, 315

[305] Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh Re-Keying: Security against Side-Channel and Fault Attacks for Low-Cost Devices. In *AFRICACRYPT*, volume 6055 of *LNCS*, pages 279–296. Springer, May 03-06 2010. Stellenbosch, South Africa. 333, 334, 336, 340

[306] Nele Mentens, Benedikt Gierlichs, and Ingrid Verbauwhede. Power and Fault Analysis Resistance in Hardware through Dynamic Reconfiguration. In *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 346–362. Springer, August 10–13 2008. Washington, D.C., USA. 221, 293

[307] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In *CHES*, volume 1965 of *LNCS*, pages 238–251. Springer-Verlag, August 17-18 2000. Worcester, MA, USA. 80, 238

[308] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX — Smartcard'99*, pages 151–162, May 10–11 1999. Chicago, Illinois, USA (Online PDF). 120, 245

[309] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Examining Smart-Card Security under the Threat of Power Analysis Attacks. *IEEE Trans. Computers*, 51(5):541–552, 2002. 77

[310] Olivier Meynard, Sylvain Guilley, Jean-Luc Danger, Yu-Ichi Hayashi, and Naofumi Homma. Identification of Information Leakage Points on a Cryptographic Device with an RSA Processor. In *IEEE EMC, Session Information Leakage*, pages 773–778, August 14-19 2011. Long Beach, CA, USA (http://www.emc2011.org). DOI: 10.1109/ISEMC.2011.6038413. 58

[311] Olivier Meynard, Sylvain Guilley, Jean-Luc Danger, and Laurent Sauvage. Far Correlation-based EMA with a precharacterized leakage model. In *DATE'10*, pages 977–980. IEEE Computer Society, March 8-12 2010. Dresden, Germany. 58

[312] Olivier Meynard, Sylvain Guilley, Denis Réal, and Jean-Luc Danger. Time Samples Correlation Attack. In *COSADE*, pages 67–72, February 24-25 2011. Darmstadt, Germany. http://cosade2011.cased.de/files/2011/cosade2011_talk7_paper.pdf. 58

[313] Olivier Meynard, Sylvain Guilley, Denis Réal, and Jean-Luc Danger. Utilisation de méthodes d'analyse fréquentielle pour l'attaque de composants cryptographiques par canaux auxiliaires, May 18 2011. http://www.lirmm.fr/journees_securite/material/j4/Meynard.pdf. 58

[314] Olivier Meynard, Denis Réal, Sylvain Guilley, Jean-Luc Danger, and Naofumi Homma. Enhancement of Simple Electro-Magnetic Attacks by Pre-characterization in Frequency Domain and Demodulation Techniques. In *DATE*. IEEE Computer Society, March 14-18 2011. Grenoble, France. 58, 79

[315] Olivier Meynard, Denis Réal, Sylvain Guilley, Florent Flament, Jean-Luc Danger, and Frédéric Valette. Characterization of the Electro-Magnetic Side Channel in Frequency Domain. In *Inscrypt (Information Security and Cryptology – 6th International Conference)*, volume 6584 of *LNCS*, pages 471–486. Springer, October 20-24 2010. Shanghai, China. DOI: 10.1007/978-3-642-21518-6_33. 58

[316] Yannick Monnet, Marc Renaudin, Régis Leveugle, Christophe Clavier, and Pascal Moitrel. Case Study of a Fault Attack on Asynchronous DES Crypto-Processors. In *FDTC*, volume 4236 of *Lecture Notes in Computer Science*, pages 88–97. Springer, October 10 2006. Yokohama, Japan. 317, 323

[317] Simon Moore, Ross Anderson, Robert Mullins, George Taylor, and Jacques J.A. Fournier. Balanced Self-Checking Asynchronous Logic for Smart Card Applications. *Journal of Microprocessors and Microsystems*, 27(9):421–430, October 2003. 106, 158

[318] Simon Moore, Robert Mullins, Paul Cunningham, Ross Anderson, and George Taylor. Improving smart card security using self-timed circuits. In *ASYNC (Asynchronous Circuits and Systems)*, pages 211– 218, April 2002. ISSN: 1522-8681, ISBN: 0-7695-1540-1j INSPEC Accession Number: 7321683. 317, 323

[319] Simon W. Moore, Ross J. Anderson, Robert D. Mullins, George S. Taylor, and Jacques J. A. Fournier. Balanced self-checking asynchronous logic for smart card applications. *Microprocessors and Microsystems*, 27(9):421–430, 2003. 290, 317

[320] Amir Moradi, Thomas Eisenbarth, Axel Poschmann, Carsten Rolfes, Christof Paar, Mohammad T. Manzuri Shalmani, and Mahmoud Salmasizadeh. Information Leakage of Flip-Flops in DPA-Resistant Logic Styles. Cryptology ePrint Archive, Report 2008/188, 2008. http://eprint.iacr.org/. 324

[321] Amir Moradi, Markus Kasper, and Christof Paar. On the Portability of Side-Channel Attacks — An Analysis of the Xilinx Virtex 4 and Virtex 5 Bitstream Encryption Mechanism. Cryptology ePrint Archive, Report 2011/391, 2011. http://eprint.iacr.org/2011/391/. 11

[322] Amir Moradi, Oliver Mischke, Christof Paar, Yang Li, Kazuo Ohta, and Kazuo Sakiyama. On the Power of Fault Sensitivity Analysis and Collision Side-Channel Attacks in a Combined Setting. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES*, volume 6917 of *LNCS*, pages 292–311. Springer, 2011. 40

[323] Elke De Mulder, Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede. Practical DPA Attacks on MDPL. In *First International Workshop on Information Forensics and Security (WIFS)*. IEEE Signal Processing Society, December 6-9 2009. London, UK. Also http://eprint.iacr.org/2009/231. 84

[324] Cédric Murdica, Sylvain Guilley, Jean-Luc Danger, Philippe Hoogvorst, and David Naccache. Same Values Power Analysis Using Special Points on Elliptic Curves. In Werner Schindler and Sorin A. Huss, editors, *COSADE*, volume 7275 of *LNCS*, pages 183–198. Springer, 2012. 58

[325] Cédric Murdica, Sylvain Guilley, and Philippe Hoogvorst. Low-Cost Countermeasure against RPA. In *CARDIS*, LNCS. Springer, November 28-30 2012. Graz, Austria. 58

[326] Radu Muresan and Stefano Gregori. Protection Circuit against Differential Power Analysis Attacks for Smart Cards. *IEEE Trans. Computers*, 57(11):1540–1549, 2008. 4

[327] Radu Muresan and Stefano Gregori. Current flattening and current sensing methods and devices. United States Patent 7716502, May 11 2010. University of Guelph, Canada. 4

[328] Chris J. Myers. *Asynchronous Circuit Design*. John Wiley & Sons, Inc., 2003. ISBN 0-471-41543-X. 31

[329] David Naccache. Finding Faults. *IEEE Security & Privacy*, 3(5):61–65, 2005. 2

[330] David Naccache. Cryptophthora. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security (2nd Ed.)*, page 284. Springer, 2011. 2

[331] Maxime Nassar, Shivam Bhasin, Jean-Luc Danger, Guillaume Duc, and Sylvain Guilley. BCDL: A high performance balanced DPL with global precharge and without early-evaluation. In *DATE'10*, pages 849–854. IEEE Computer Society, March 8-12 2010. Dresden, Germany. 33, 58, 81, 323

[332] Maxime Nassar, Sylvain Guilley, and Jean-Luc Danger. Formal Analysis of the Entropy / Security Trade-off in First-Order Masking Countermeasures against Side-Channel Attacks. In *INDOCRYPT*, volume 7107 of *LNCS*, pages 22–39. Springer, December 11-14 2011. Chennai, Tamil Nadu, India. DOI: 10.1007/978-3-642-25578-6_4. vi, 19, 42, 58, 205

[333] Maxime Nassar, Youssef Souissi, Sylvain Guilley, and Jean-Luc Danger. The "Rank Correction" Technique to Improve Side-Channel Attacks. In *CryptArchi*, Gif-sur-Yvette, France, June 27-30 2010. Gif-sur-Yvette, France; (abstract). 58

[334] Maxime Nassar, Youssef Souissi, Sylvain Guilley, and Jean-Luc Danger. RSM: a Small and Fast Countermeasure for AES, Secure against First- and Second-order Zero-Offset SCAs. In *DATE*, pages 1173–1178, March 12-16 2012. Dresden, Germany. (TRACK A: "Application Design", TOPIC A5: "Secure Systems"). 19, 42, 58, 71, 208

[335] Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. An Integrated Validation Environment for Differential Power Analysis. In *DELTA*, pages 527–532, Los Alamitos, CA, USA, 2008. IEEE Computer Society. 187

[336] NIST/ITL/CSD. Data Encryption Standard. FIPS PUB 46-3, Oct 1999. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf. 1, 124, 318

[337] NIST/ITL/CSD. Advanced Encryption Standard (AES). FIPS PUB 197, Nov 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. 1, 157, 158, 302, 306, 318

[338] NIST/ITL/CSD. Secure Hash Algorithm (SHA). FIPS PUB 180-2, Nov 2001. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf. 1

[339] Karsten Nohl, David Evans, Starbug, and Henryk Plötz. Reverse-Engineering a Cryptographic RFID Tag. In Paul C. van Oorschot, editor, *USENIX Security Symposium*, pages 185–194. USENIX Association, July 28th – August 1st 2008. San Jose, CA, USA. 253

[340] Karsten Nohl, David Evans Starbug, and Henryk Plötz. Reverse-Engineering a Cryptographic RFID Tag. In *USENIX Security Symposium*, pages 185–193, July 31 2008. San Jose, CA, USA (Online HTML). 251

[341] Karsten Nohl, Erik Tews, and Ralf-Philipp Weinmann. Cryptanalysis of the DECT Standard Cipher. In *FSE*, Lecture Notes in Computer Science. Springer, February 7-10 2010. Seoul, South Korea. 253

[342] Roman Novak. Side-Channel Attack on Substitution Blocks. In *ACNS*, volume 2846 of *LNCS*, pages 307–318. Springer, October 2003. Kunming, China. 250, 252

[343] Roman Novak. Side-Channel Based Reverse Engineering of Secret Algorithms. In Baldomir Zajc, editor, *Proceedings of the Twelfth International Electrotechnical and Computer Science Conference (ERK 2003)*, pages 445–448, Ljubljana, Slovenia, September 25-26 2003. Slovenska sekcija IEEE. 252

[344] Roman Novak. Sign-Based Differential Power Analysis. In *WISA*, volume 2908 of *LNCS*, pages 203–216. Springer, 2003. Jeju Island, Korea. 250, 252

[345] Elisabeth Oswald. http://opensca.sourceforge.net/, U. of Bristol, UK, 2010. 78

[346] Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller, and Vincent Rijmen. A Side-Channel Analysis Resistant Description of the AES S-box. In LNCS, editor, *Proceedings of FSE'05*, volume 3557 of *LNCS*, pages 413–423. Springer, February 2005. Paris, France. 146

[347] Christof Paar, Thomas Eisenbarth, Markus Kasper, Timo Kasper, and Amir Moradi. KeeLoq and Side-Channel Analysis — Evolution of an Attack. In Luca Breveglieri, Israel Koren, David Naccache, Elisabeth Oswald, and Jean-Pierre Seifert, editors, *FDTC*, pages 65–69. IEEE Computer Society, 2009. 3

[348] Christof Paar, Thomas Eisenbarth, Markus Kasper, Timo Kasper, and Amir Moradi. KeeLoq and Side-Channel Analysis-Evolution of an Attack. In *FDTC*, pages 65–69. IEEE, 6 September 2009. Lausanne, Switzerland. 76

[349] Renaud Pacalet and Sylvain Guilley. Asynchronisme, sécurité et consommation. In *ECoFac 2006: école thématique ECoFac "Conception faible consommation de système temps réel"*. 3 – 7 avril 2006, Nice, France, (Online PDF). 58

[350] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, May 2-6 1999. Prague, Czech Republic. 313

[351] Manuel San Pedro, Soos Mate, and Sylvain Guilley. FIRE: Fault Injection for Reverse Engineering. In LNCS, editor, *WISTP: Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing*, volume 6633 of *LNCS*, pages 280–293. Springer, June 1-3 2011. Heraklion, Greece. DOI: 10.1007/978-3-642-21040-2_20. 58

[352] Éric Peeters. *Towards Security Limits of Embedded Hardware Devices: from Practice to Theory*. PhD thesis, Université catholique de Louvain, November 2006. 185

[353] Éric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Power and electromagnetic analysis: Improved model, consequences and comparisons. *Integration, The VLSI Journal, special issue on* "Embedded Cryptographic Hardware", 40:52–60, January 2007. DOI: 10.1016/j.vlsi.2005.12.013. 120

[354] Marcel J.M. Pelgrom, Aad C.J. Duinmaijer, and Anton P.G. Welbers. Matching properties of MOS transistors. *IEEE Journal of Solid State Circuits*, 24(5):1433–1439, 1989. DOI: 10.1109/JSSC.1989.572629. 94

[355] Krzysztof Pietrzak. A Leakage-Resilient Mode of Operation. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 462–482. Springer, April 26-30 2009. Cologne, Germany. 331

[356] Gilles Piret. A Note on the Plaintexts Choice in Power Analysis Attacks. Technical Report from the École Normale Supérieure (ENS), France, November 2005. `http://www.di.ens.fr/~piret/publ/power.pdf`. 134, 138

[357] Gilles Piret and Jean-Jacques Quisquater. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad. In *CHES*, volume 2779 of *LNCS*, pages 77–88. Springer, September 2003. Cologne, Germany. 271, 292, 317, 318

[358] Thomas Popp, Mario Kirschbaum, Thomas Zefferer, and Stefan Mangard. Evaluation of the Masked Logic Style MDPL on a Prototype Chip. In *CHES*, volume 4727 of *LNCS*, pages 81–94. Springer, Sept 2007. Vienna, Austria. 127, 146, 295, 323

[359] Thomas Popp and Stefan Mangard. Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. In *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 172–186. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK. 104, 146, 157, 160, 295

[360] Axel Poschmann. *Lightweight Cryptography – Cryptographic Engineering for a Pervasive World*. PhD thesis, Ruhr-University Bochum, February 2009. Referees: Prof. Christof Paar and Dr. Matthew J.B. Robshaw (Orange Labs, France Telekom). See also the Cryptology ePrint Archive, report 2009/516. 341

[361] Viktor K. Prasanna, Jürgen Becker, and René Cumplido, editors. *ReConFig'10: 2010 International Conference on Reconfigurable Computing and FPGAs, Cancun, Quintana Roo, Mexico, 13-15 December 2010, Proceedings*. IEEE Computer Society, 2010. 348, 369

[362] Emmanuel Prouff. DPA Attacks and S-Boxes. In *FSE*, volume 3557 of *LNCS*, pages 424–441. Springer-Verlag, february 2005. Paris, France. 165

[363] Emmanuel Prouff and Matthieu Rivain. A Generic Method for Secure SBox Implementation. In Sehun Kim, Moti Yung, and Hyung-Woo Lee, editors, *WISA*, volume 4867 of *Lecture Notes in Computer Science*, pages 227–244. Springer, 2007. 207

[364] Emmanuel Prouff and Matthieu Rivain. Theoretical and Practical Aspects of Mutual Information Based Side Channel Analysis. In Springer, editor, *ACNS*, volume 5536 of *LNCS*, pages 499–518, June 2-5 2009. Paris-Rocquencourt, France. 10, 28, 200

[365] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009. 21, 43, 211, 246

[366] Emmanuel Prouff and Thomas Roche. Attack on a Higher-Order Masking of the AES Based on Homographic Functions. In Guang Gong and Kishan Chand Gupta, editors, *INDOCRYPT*, volume 6498 of *Lecture Notes in Computer Science*, pages 262–281. Springer, 2010. 80

[367] NIST FIPS (Federal Information Processing Standards) publication 140-2. Security Requirements for Cryptographic Modules. page 69, May 25 2001. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf. 10, 326, 327

[368] NIST FIPS (Federal Information Processing Standards) publication 140-3. Security Requirements for Cryptographic Modules (Draft, Revised). page 63, 09/11 2009. http://csrc.nist.gov/groups/ST/FIPS140_3/. 326, 327

[369] Jean-Jacques Quisquater and David Samyde. Radio frequency attacks. In Henk C. A. van Tilborg, editor, *Encyclopedia of Cryptography and Security*, pages 503–509. Springer, 2005. 297

[370] Jean-Jacques Quisquater and François-Xavier Standaert. Physically Secure Cryptographic Computations: From Micro to Nano Electronic Devices. In *DSN, Workshop on Dependable and Secure Nanocomputing (WDSN)*. IEEE Computer Society, June 28 2007. Invited Talk, 2 pages, Edinburgh, UK. 200

[371] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. *Digital Integrated Circuits*. Prentice Hall, 2003. ISBN-10: 0130909963, 761 pages. 125

[372] A. Raghunathan, S. Dey, and N.K. Jha. High-level macro-modeling and estimation techniques for switching activity and power consumption. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 11(4):538 –557, aug. 2003. 32

[373] Alin Razafindraibe, Michel Robert, Marc Renaudin, and Philippe Maurine. A Method to Design Compact Dual-rail Asynchronous Primitives. In *PATMOS*, pages 571–580, 2005. http://dx.doi.org/10.1007/11556930_58. 104

[374] Denis Réal, Vivien Dubois, Anne-Marie Guilloux, Frédéric Valette, and M'hamed Drissi. SCARE of an Unknown Hardware Feistel Implementation. In *CARDIS*, volume 5189 of *LNCS*, pages 218–227. Springer, 2008. London, UK. 250, 252

[375] Denis Réal, Frédéric Valette, and M'hamed Drissi. Enhancing correlation electromagnetic attack using planar near-field cartography. In *DATE*, pages 628–633. IEEE, April 20-24 2009. Nice, France. 235

[376] Christian Rechberger and Elisabeth Oswald. Practical Template Attacks. In *WISA*, volume 3325 of *LNCS*, pages 443–457. Springer, August 23-25 2004. Jeju Island, Korea. 120

[377] Robert Redelmeier. `cpuburn`, CPU testing utilities, June 16 2001. Software available on-line: http://pages.sbcglobal.net/redelm/ under GNU Public Licence. 306

[378] Francesco Regazzoni, Stéphane Badel, Thomas Eisenbarth, Johann Großschädl, Axel Poschmann, Zeynep Toprak, Marco Macchetti, Laura Pozzi, Christof Paar, Yusuf Leblebici, and Paolo Ienne. A Simulation-Based Methodology for Evaluating DPA-Resistance of Cryptographic Functional Units with Application to CMOS and MCML Technologies. In *International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS IC 07)*, July 2007. Samos, Greece. 125

[379] Francesco Regazzoni, Alessandro Cevrero, François-Xavier Standaert, Stéphane Badel, Theo Kluter, Philip Brisk, Yusuf Leblebici, and Paolo Ienne. A Design Flow and Evaluation Framework for DPA-Resistant Instruction Set Extensions. In *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 205–219. Springer, 6-9 September 2009. Lausanne, Switzerland. 183

[380] Francesco Regazzoni, Thomas Eisenbarth, Johann Großschädl, Luca Breveglieri, Paolo Ienne, Israel Koren, and Christof Paar. Power Attacks Resistance of Cryptographic S-Boxes with Added Error Detection Circuits. In *DFT*, pages 508–516. IEEE Computer Society, September 26-28 2007. Rome, Italy. 323

[381] Francesco Regazzoni, Yi Wang, and François-Xavier Standaert. FPGA Implementations of the AES Masked Against Power Analysis Attacks. In *COSADE*, pages 56–66, February 2011. Darmstadt, Germany. 15

[382] Vincent Rijmen. Efficient Implementation of the Rijndael S-box. Informal communication. 158

[383] Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *LNCS*, pages 413–427. Springer, 2010. 207

[384] Ronald L. Rivest. All-or-Nothing Encryption and the Package Transform. In *FSE*, volume 1267 of *LNCS*, pages 210–218. Springer, January 20-22 1997. Haifa, Israel. 335

[385] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. 306

[386] Bruno Robisson and Pascal Manet. Differential Behavioral Analysis. In *CHES*, volume 4727 of *LNCS*, pages 413–426. Springer, September 10-13 2007. Vienna, Austria. 289, 308, 316

[387] Thomas Roche and Cédric Tavernier. Multi-Linear cryptanalysis in Power Analysis Attacks: MLPA. In *Western European Workshop on Research in Cryptology, WEWoRC 2009*, July 7-9 2009. Graz, Austria. 194

[388] Atri Rudra, Pradeep K. Dubey, Charanjit S. Jutla, Vijay Kumar, Josyula R. Rao, and Pankaj Rohatgi. Efficient Rijndael Encryption Implementation with Composite Field Arithmetic. In *CHES*, volume 2162 of *LNCS*, pages 171–184, London, UK, May 2001. Springer-Verlag. 158

[389] Andrew R. Runnalls. Kullback-Leibler Approach to Gaussian Mixture Reduction. *Aerospace and Electronic Systems, IEEE Transactions on*, 43(3):989 –999, july 2007. 25, 49

[390] Minoru Saeki and Daisuke Suzuki. Security Evaluations of MRSL and DRSL Considering Signal Delays. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E91-A(1):176–183, 2008. DOI: 10.1093/ietfec/e91-a.1.176. 38, 294

[391] Akashi Satoh. Side-channel Attack Standard Evaluation Board, SASEBO. Project of the AIST – RCIS (Research Center for Information Security), http://www.risec.aist.go.jp/project/sasebo/. 187, 242

[392] Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh. A Compact Rijndael Hardware Architecture with S-Box Optimization. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 239–254. Springer, 2001. 15

[393] Laurent Sauvage, Sylvain Guilley, Jean-Luc Danger, Naofumi Homma, and Yu-Ichi Hayashi. Practical Results of EM Cartography on a FPGA-based RSA Hardware Implementation. In *IEEE EMC, Session Information Leakage*, pages 768–772, August 14-19 2011. Long Beach, CA, USA (http://www.emc2011.org). DOI: 10.1109/ISEMC.2011.6038412. 58

[394] Laurent Sauvage, Sylvain Guilley, Jean-Luc Danger, Naofumi Homma, and Yu-Ichi Hayashi. A Fault Model for Conducted Intentional ElectroMagnetic Interferences. In *Electromagnetic Compatibility (EMC), 2012 IEEE International Symposium on*, pages 788–793, August 5-10 2012. Pittsburgh, PA, USA (http://2012emc.org/). DOI: 10.1109/ISEMC.2012.6351664. 58

[395] Laurent Sauvage, Sylvain Guilley, Jean-Luc Danger, Yves Mathieu, and Maxime Nassar. Successful Attack on an FPGA-based WDDL DES Cryptoprocessor Without Place and Route Constraints. In *DATE*, pages 640–645, Nice, France, apr 2009. IEEE Computer Society. 58, 319, 320

[396] Laurent Sauvage, Sylvain Guilley, Florent Flament, Jean-Luc Danger, and Yves Mathieu. Cross-correlation Cartography. In *ReConFig*, pages 268–273. IEEE Computer Society, December 13–15 2010. Cancún, Quintana Roo, México. DOI: 10.1109/ReConFig.2010.75. 20, 58

[397] Laurent Sauvage, Sylvain Guilley, Florent Flament, Jean-Luc Danger, and Yves Mathieu. Blind Cartography for Side Channel Attacks – Cross-correlation Cartography. *International Journal of Reconfigurable Computing (IJRC)*, page 9, 2012. Article ID 360242. DOI: 10.1155/2012/360242. 58

[398] Laurent Sauvage, Sylvain Guilley, and Yves Mathieu. ElectroMagnetic Radiations of FPGAs: High Spatial Resolution Cartography and Attack of a Cryptographic Module. *ACM Trans. Reconfigurable Technol. Syst.*, 2(1):1–24, March 2009. Full text in http://hal.archives-ouvertes.fr/hal-00319164/en/. 4

[399] Laurent Sauvage, Sylvain Guilley, and Yves Mathieu. ElectroMagnetic Radiations of FPGAs: High Spatial Resolution Cartography and Attack of a Cryptographic Module. *TRETS (ACM Transactions on Reconfigurable Technologies and Systems)*, jan 2009. 58

[400] Laurent Sauvage, Olivier Meynard, Sylvain Guilley, and Jean-Luc Danger. ElectroMagnetic Attacks Case Studies on Non-Protected and Protected Cryptographic Hardware Accelerators. In *IEEE EMC, Special session #4 on Modeling/Simulation Validation and use of FSV*, July 25-30 2010. Fort Lauderdale, Florida, USA (http://emc2010.org/). 58

[401] Laurent Sauvage, Maxime Nassar, Sylvain Guilley, Florent Flament, Jean-Luc Danger, and Yves Mathieu. DPL on Stratix II FPGA: What to Expect? In *ReConFig*, pages 243–248. IEEE Computer Society, December 9–11 2009. Cancún, Quintana Roo, México, DOI: 10.1109/ReConFig.2009.58. 34, 58

[402] Laurent Sauvage, Maxime Nassar, Sylvain Guilley, Florent Flament, Jean-Luc Danger, and Yves Mathieu. Exploiting Dual-Output Programmable Blocks to Balance Secure Dual-Rail Logics. *International Journal of Reconfigurable Computing (IJRC)*, page 12, 2010. DOI: 10.1155/2010/375245. 34, 58

[403] Amitabh Saxena, Brecht Wyseur, and Bart Preneel. Towards Security Notions for White-Box Cryptography. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio Agostino Ardagna, editors, *ISC*, volume 5735 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2009. 13

[404] SCARD European sixth framework programme (FP6) project website: http://www.scard-project.eu. 127, 146

[405] Patrick Schaumont and Kris Tiri. Masking and Dual Rail Logic Don't Add Up. In *CHES*, volume 4727 of *LNCS*, pages 95–106. Springer, September 10-13 2007. Vienna, Austria. 84, 157

[406] Werner Schindler. Advanced stochastic methods in side channel analysis on block ciphers in the presence of masking. *Journal of Mathematical Cryptology*, 2(3):291–310, October 2008. ISSN (Online) 1862-2984, ISSN (Print) 1862-2976, DOI: 10.1515/JMC.2008.013. 77, 210

[407] Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In LNCS, editor, *CHES*, volume 3659 of *LNCS*, pages 30–46. Springer, Sept 2005. Edinburgh, Scotland, UK. 77, 258

[408] Martin Schobert. GNU software DEGATE. Webpage: http://www.degate.org/. 111

[409] Pete Sedcole and Peter Y. K. Cheung. Within-die delay variability in 90nm FPGAs and beyond. In IEEE, editor, *ICFPT*, pages 97–104, dec 2006. Bangkok, Thailand. DOI: 10.1109/FPT.2006.270300. 94

[410] Nidhal Selmane, Shivam Bhasin, Sylvain Guilley, and Jean-Luc Danger. Security evaluation of application-specific integrated circuits and field programmable gate arrays against setup time violation attacks. *IET Information Security*, 5(4):181–190, December 2011. DOI: 10.1049/iet-ifs.2010.0238. 58, 288

[411] Nidhal Selmane, Shivam Bhasin, Sylvain Guilley, Tarik Graba, and Jean-Luc Danger. WDDL is Protected Against Setup Time Violation Attacks. In *FDTC*, pages 73–83. IEEE Computer Society, September 6th 2009. In conjunction with CHES'09, Lausanne, Switzerland. DOI: 10.1109/FDTC.2009.40; Online version: http://hal.archives-ouvertes.fr/hal-00410135/en/. viii, 39, 58, 269, 297, 299, 302, 317

[412] Nidhal Selmane, Sylvain Guilley, and Jean-Luc Danger. Setup Time Violation Attacks on AES. In *EDCC, The seventh European Dependable Computing Conference*, pages 91–96, Kaunas, Lithuania, May 7-9 2008. ISBN: 978-0-7695-3138-0, DOI: 10.1109/EDCC-7.2008.11. 58, 201, 270, 280, 306

[413] Shaunak Shah, Rajesh Velegalati, Jens-Peter Kaps, and David Hwang. Investigation of DPA Resistance of Block RAMs in Cryptographic Implementations on FPGAs. In Prasanna et al. [361], pages 274–279. 13

[414] M. Shams, J.C. Ebergen, and M.I. Elmasry. Modeling and comparing CMOS implementations of the C-Element. *IEEE Transactions on VLSI Systems*, 6(4):563–567, December 1998. 107, 128, 160

[415] Carsten Sinz. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In Peter van Beek, editor, *CP*, volume 3709 of *Lecture Notes in Computer Science*, pages 827–831. Springer, 2005. 218

[416] Sergei P. Skorobogatov. Research project: developing new technology for effective side-channel analysis. `http://www.cl.cam.ac.uk/~sps32/qvl_proj.html`, accessed September 12th, 2011. 3

[417] Rafael Soares, Ney Calazans, Victor Lomné, Philippe Maurine, Lionel Torres, and Michel Robert. Evaluating the robustness of secure triple track logic through prototyping. In *SBCCI'08: Proceedings of the 21st annual symposium on Integrated circuits and system design*, pages 193–198, New York, NY, USA, September 1-4 2008. ACM. 295, 323

[418] Rafael Soares, Ney Calazans, Victor Lomne, Thomas Ordas, Philippe Maurine, Lionel Torres, and Michel Robert. Evaluation on FPGA of Triple Rail Logic Robustness against DPA and DEMA. In *DATE, track A4 (*Secure embedded implementations*)*, pages 634–639. IEEE, April 20–24 2009. Nice, France. 323

[419] Danil Sokolov, Julian Murphy, Alexander Bystrov, and Alex Yakovlev. Design and Analysis of Dual-Rail Circuits for Security Applications. *IEEE Trans. Comput.*, 54(4):449–460, 2005. 318

[420] Danil Sokolov, Julian Murphy, Alexandre V. Bystrov, and Alexandre Yakovlev. Improving the Security of Dual-Rail Circuits. In *CHES*, volume 3156 of *LNCS*, pages 282–297. Springer, August 11-13 2004. Cambridge, MA, USA. 31, 106

[421] Mate Soos. SAT-solver "`cryptominisat`", Version 2.9.0, January 20 2011. `https://gforge.inria.fr/projects/cryptominisat`. 218

[422] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT Solvers to Cryptographic Problems. In Oliver Kullmann, editor, *SAT*, volume 5584 of *Lecture Notes in Computer Science*, pages 244–257. Springer, 2009. 218

[423] Youssef Souissi, Shivam Bhasin, Sylvain Guilley, Maxime Nassar, and Jean-Luc Danger. Towards Different Flavors of Combined Side Channel Attacks. In *CT-RSA*, volume 7178 of *LNCS*, pages 245–259. Springer, February 27 – March 2 2012. San Francisco, CA, USA. DOI: 10.1007/978-3-642-27954-6_16. 58, 71, 77

[424] Youssef Souissi, Jean-Luc Danger, Sylvain Guilley, Shivam Bhasin, and Maxime Nassar. Common Framework to Evaluate Modern Embedded Systems against Side-Channel Attacks. In *HST (International Conference on Technologies for Homeland Security)*, IEEE, pages 86–91, November 15-17 2011. Westin Hotel, Waltham, MA, USA. DOI: 10.1109/THS.2011.6107852. 11, 58

[425] Youssef Souissi, Jean-Luc Danger, Sylvain Guilley, Shivam Bhasin, and Maxime Nassar. Embedded Systems Security: An Evaluation Methodology Against Side Channel Attacks. In *DASIP*, IEEE Signal Processing Society, pages 230–237, November 2-4 2011. Tampere, Finland. DOI: 10.1109/DASIP.2011.6136885. 11, 58

[426] Youssef Souissi, Jean-Luc Danger, Sami Mekki, Sylvain Guilley, and Maxime Nassar. Techniques for electromagnetic attacks enhancement. In *DTIS (Design & Technologies of Integrated Systems)*, IEEE, pages 1–6. IEEE, March 23-25 2010. Hammamet, Tunisia; DOI: 10.1109/DTIS.2010.5487590. 58

[427] Youssef Souissi, Nicolas Debande, Sami Mekki, Sylvain Guilley, Ali Maalaoui, and Jean-Luc Danger. On the Optimality of Correlation Power Attack on Embedded Cryptographic Systems. In

Ioannis G. Askoxylakis, Henrich Christopher Pöhls, and Joachim Posegga, editors, *WISTP*, volume 7322 of *Lecture Notes in Computer Science*, pages 169–178. Springer, June 20-22 2012. 58

[428] Youssef Souissi, Moulay Aziz Elaabid, Jean-Luc Danger, Sylvain Guilley, and Nicolas Debande. Novel Applications of Wavelet Transforms based Side-Channel Analysis, September 26-27 2011. Non-Invasive Attack Testing Workshop (NIAT 2011), co-organized by NIST & AIST. Todai-ji Cultural Center, Nara, Japan. (PDF). 58

[429] Youssef Souissi, Sylvain Guilley, Jean-Luc Danger, Guillaume Duc, and Sami Mekki. Improvement of power analysis attacks using Kalman filter. In *ICASSP*, IEEE Signal Processing Society, pages 1778–1781. IEEE, March 14-19 2010. Dallas, TX, USA; DOI: 10.1109/ICASSP.2010.5495428. 58

[430] Youssef Souissi, Maxime Nassar, Sylvain Guilley, Jean-Luc Danger, and Florent Flament. First Principal Components Analysis: A New Side Channel Distinguisher. In Kyung Hyune Rhee and DaeHun Nyang, editors, *ICISC*, volume 6829 of *Lecture Notes in Computer Science*, pages 407–419. Springer, 2010. 28

[431] Youssef Souissi, Maxime Nassar, Sylvain Guilley, Jean-Luc Danger, and Florent Flament. First Principal Components Analysis: A New Side Channel Distinguisher. In *ICISC*, LNCS. Springer, December 1-3 2010. Seoul, Korea. 58, 77

[432] François-Xavier Standaert and Cédric Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, August 10–13 2008. Washington, D.C., USA. 235, 242

[433] François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede. Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices. In *ICISC*, volume 5461 of *LNCS*, pages 253–267. Springer, December 3-5 2008. Seoul, Korea. 23, 30, 77, 234, 238

[434] François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany. 10, 37, 81, 206, 208, 209, 210, 236, 308, 331

[435] François-Xavier Standaert, Sıddıka Berna Örs, and Bart Preneel. Power Analysis of an FPGA: Implementation of Rijndael: Is Pipelining a DPA Countermeasure? In *CHES*, volume 3156 of *LNCS*, pages 30–44. Springer-Verlag, August 11-13 2004. Cambridge (Boston), MA, USA. 24, 192, 258

[436] François-Xavier Standaert, Éric Peeters, François Macé, and Jean-Jacques Quisquater. Updates on the Security of FPGAs Against Power Analysis Attacks. In *ARC*, volume 3985 of *LNCS*, pages 335–346. Springer-Verlag, March 2006. Delft, The Netherlands. 208

[437] François-Xavier Standaert, Éric Peeters, Gaël Rouvroy, and Jean-Jacques Quisquater. An Overview of Power Analysis Attacks Against Field Programmable Gate Arrays. *Proceedings of the IEEE*, 94(2):383–394, February 2006. (Invited Paper). 200, 259

[438] François-Xavier Standaert, Gaël Rouvroy, and Jean-Jacques Quisquater. FPGA Implementations of the DES and Triple-DES Masked Against Power Analysis Attacks. In *FPL*. IEEE, August 2006. Madrid, Spain. 207

[439] Daisuke Suzuki and Minoru Saeki. Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. In *CHES*, volume 4249 of *LNCS*, pages 255–269. Springer, October 10-13 2006. Yokohama, Japan. http://dx.doi.org/10.1007/11894063_21. 86, 104, 107, 114, 121, 126, 158, 270, 319

[440] Daisuke Suzuki and Minoru Saeki. An Analysis of Leakage Factors for Dual-Rail Pre-Charge Logic Style. *IEICE Transactions*, 91-A(1):184–192, 2008. 38

[441] Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. Random Switching Logic: A Countermeasure against DPA based on Transition Probability, 2004. http://eprint.iacr.org/2004/346. 104, 294

[442] Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. Random Switching Logic: A New Countermeasure against DPA and Second-Order DPA at the Logic Level. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E90-A(1):160–168, 2007. 294

[443] Christopher Tarnovsky. How to Reverse-Engineer a Satellite TV Smart Card, 2010. Online video: http://www.youtube.com/watch?v=tnY7UVyaFiQ. 9

[444] Mohammad Tehranipoor and Cliff Wang, editors. *Introduction to Hardware Security and Trust*. Springer, 2012. ISBN 978-1-4419-8079-3. 2

[445] TELECOM ParisTech SEN research group. DPA Contest (1st edition), 2008–2009. http://www.DPAcontest.org/. 192, 234, 238, 259, 292

[446] TELECOM ParisTech SEN research group. DPA Contest (2nd edition), 2009–2010. http://www.DPAcontest.org/v2/. 11

[447] TELECOM ParisTech SEN research group (contact@dpacontest.org). DPA Contests, 2008–2011. http://www.DPAcontest.org/home/. 74, 78

[448] The "Xilinx TMR Tool" (*TMP is short for Triple Module Redundancy*). Features description at this web page:. http://www.xilinx.com/ise/optional_prod/tmrtool.htm. 325

[449] Stefan Tillich, Martin Feldhofer, and Johann Großschädl. Area, Delay, and Power Characteristics of Standard-Cell Implementations of the AES S-Box. In *SAMOS*, volume 4017 of *LNCS*, pages 457–466. Springer-Verlag, July 17-20 2006. Samos, Greece. 158

[450] Stefan Tillich, Martin Feldhofer, Thomas Popp, and Johann Großschädl. Area, delay, and power characteristics of standard-cell implementations of the AES S-Box. *J. Signal Process. Syst.*, 50(2):251–261, 2008. 158

[451] Kris Tiri. Side-Channel Attack Pitfalls. In *44th Design Automation Conference (DAC)*, pages 15–20, June 4 & 8 2007. San Diego, California, USA. 132, 133

[452] Kris Tiri, Moonmoon Akmal, and Ingrid Verbauwhede. A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. In *European Solid-State Circuits Conference (ESSCIRC)*, pages 403–406, September 2002. Florence, Italy, http://citeseer.ist.psu.edu/tiri02dynamic.html. 94, 104, 125, 270

[453] Kris Tiri, David Hwang, Alireza Hodjat, Bo-Cheng Lai, Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede. A side-channel leakage free coprocessor IC in 0.18 μm CMOS for Embedded AES-based Cryptographic and Biometric Processing. In *DAC*, pages 222–227. ACM, June 13-17 2005. San Diego, CA, USA. 323, 324

[454] Kris Tiri, Davis Hwang, Alireza Hodjat, Bo-Cheng Lai, Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede. Prototype IC with WDDL and Differential Routing – DPA Resistance Assessment. In LNCS, editor, *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 354–365. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK. 127, 146

[455] Kris Tiri and Patrick Schaumont. Changing the odds against Masked Logic. In *13th Annual Workshop on Selected Areas in Cryptography*, volume 4356 of *LNCS*, pages 134–146. Springer, August 17 & 18 2006. Montreal, Canada. 104

[456] Kris Tiri and Ingrid Verbauwhede. A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In *DATE'04*, pages 246–251. IEEE Computer Society, February 2004. Paris, France. DOI: 10.1109/DATE.2004.1268856. 93, 104, 106, 114, 121, 125, 157, 158, 192, 270, 271, 294, 295, 319, 322

[457] Kris Tiri and Ingrid Verbauwhede. Place and Route for Secure Standard Cell Design. In Kluwer, editor, *Proceedings of WCC / CARDIS*, pages 143–158, Aug 2004. Toulouse, France. 33, 93, 122, 128, 185, 294, 319

[458] Kris Tiri and Ingrid Verbauwhede. Secure Logic Synthesis. In *FPL*, volume 3203 of *LNCS*, pages 1052–1056. Springer, August 30 – September 1 2004. Leuven, Belgium. 122, 270

[459] Kris Tiri and Ingrid Verbauwhede. Synthesis of Secure FPGA Implementations. In *International Workshop on Logic and Synthesis (IWLS'04)*, pages 224–231, June 2004. http://www.ee.ucla.edu/~tiri/files/iwls2004.pdf. 122, 270

[460] Kris Tiri and Ingrid Verbauwhede. A VLSI Design Flow for Secure Side-Channel Attack Resistant ICs. In *DATE*, pages 58–63. IEEE Computer Society, 2005. http://dx.doi.org/10.1109/DATE.2005.44. 114

[461] G. Torrens, B. Alorda, S. Barceló, J. L. Rosselló, S. Bota, and J. Segura. An SRAM SEU Hardening Technique for Multi-Vt Nanometric CMOS Technologies. In *DCIS*, November 12–14 2008. ISBN: 978-2-84813-124-5, Grenoble, France. 285

[462] Gabriel Torrens, Bartomeu Alorda, Salvador Barceló, José Luis Rosselló, Sebastiàn A. Bota, and Jaume Segura. Design Hardening of Nanometer SRAMs Through Transistor Width Modulation and Multi-Vt Combination. *IEEE Trans. on Circuits and Systems*, 57-II(4):280–284, 2010. ISSN: 1549-7747. 285

[463] Michael Tunstall and Debdeep Mukhopadhyay. Differential Fault Analysis of the Advanced Encryption Standard using a Single Fault. Report 2009/575, 2009. http://eprint.iacr.org/2009/575, to appear in the proceedings of WISTP 2011 (Springer LNCS, vol. 6633, Heraklion, Greece). 306, 318, 336

[464] Haleh Vahedi, Stefano Gregori, and Radu Muresan. The effectiveness of a current flattening circuit as countermeasure against DPA attacks. *Microelectronics Journal*, 42(1):180 – 187, 2011. 4

[465] Rajesh Velegalati and Jens-Peter Kaps. Improving security of SDDL designs through interleaved placement on Xilinx FPGAs. In *Field Programmable Logic and Applications, FPL 2011*, pages 506–511. IEEE, September 2011. Chania, Greece. DOI: 10.1109/FPL.2011.100. 34

[466] Dennis Vermoen, Marc F. Witteman, and Georgi Gaydadjiev. Reverse Engineering Java Card Applets Using Power Analysis. In Damien Sauveron, Constantinos Markantonakis, Angelos Bilas, and Jean-Jacques Quisquater, editors, *WISTP*, volume 4462 of *Lecture Notes in Computer Science*, pages 138–149. Springer, may 8-11 2007. Heraklion, Greece. 250, 252

[467] Olli Vertanen. Java Type Confusion and Fault Attacks. In *FTDC*, volume 4236 of *LNCS*, pages 237–251. Springer, 2006. DOI: 10.1007/11889700, ISSN 0302-9743 (Print) 1611-3349 (Online), ISBN 978-3-540-46250-7. 306

[468] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Mutual Information Analysis: How, When and Why? In *CHES*, volume 5747 of *LNCS*, pages 429–443. Springer, September 6-9 2009. Lausanne, Switzerland. 200, 209, 234

[469] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Adaptive Chosen-Message Side-Channel Attacks. In Jianying Zhou and Moti Yung, editors, *ACNS*, volume 6123 of *Lecture Notes in Computer Science*, pages 186–199, 2010. 209

[470] Jason Waddle and David Wagner. Towards Efficient Second-Order Power Analysis. In *CHES*, volume 3156 of *LNCS*, pages 1–15. Springer, 2004. Cambridge, MA, USA. 209

[471] Jason Waddle and David Wagner. Fault Attacks on Dual-Rail Encoded Systems. In *ACSAC*, pages 483–494. IEEE Computer Society, 2005. 40

[472] Neil H.E. Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison Wesley, 2004. 3rd edition (May 11, 2004), ISBN: 0321149017. 3

[473] Carolyn Whitnall and Elisabeth Oswald. A Comprehensive Evaluation of Mutual Information Analysis Using a Fair Evaluation Framework. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 316–334. Springer, 2011. 12

[474] Carolyn Whitnall, Elisabeth Oswald, and Luke Mather. An Exploration of the Kolmogorov-Smirnov Test as a Competitor to Mutual Information Analysis. In Emmanuel Prouff, editor, *CARDIS*, volume 7079 of *Lecture Notes in Computer Science*, pages 234–251. Springer, 2011. 77

[475] Johannes Wolkerstorfer, Elisabeth Oswald, and Mario Lamberger. An ASIC Implementation of the AES SBoxes. In Bart Preneel, editor, *CT-RSA*, volume 2271 of *Lecture Notes in Computer Science*, pages 67–78. Springer, 2002. 158, 270

[476] Qing Xu, Marcia B. Costa e Silva, Jean-Luc Danger, Sylvain Guilley, Patrick Bellot, Philippe Gallion, and Francisco J. Mendieta. Towards Quantum Key Distribution System using Homodyne Detection with Differential Time-Multiplexed Reference. pages 158–165. RIVF'07 – http://www.rivf.org/, March 05–09 2007, Hanoi, Viet Nam. DOI: 10.1109/RIVF.2007.369151. 58

[477] Sung-Ming Yen and Marc Joye. Checking Before Output May Not Be Enough Against Fault-Based Cryptanalysis. *IEEE Trans. Computers*, 49(9):967–970, 2000. DOI: 10.1109/12.869328. 41, 79, 308, 309, 312, 316

[478] Sung-Ming Yen, Seungjoo Kim, Seongan Lim, and Sang-Jae Moon. RSA Speedup with Chinese Remainder Theorem Immune against Hardware Fault Cryptanalysis. *IEEE Trans. Computers*, 52(4):461–472, 2003. DOI: 10.1109/TC.2003.1190587. 311

[479] Pengyuan Yu and Patrick Schaumont. Secure FPGA circuits using controlled placement and routing. In *CODES+ISSS'07: Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis*, pages 45–50, New York, NY, USA, 2007. ACM. 34

[480] Zhongchuan C. Yu, Stephen B. Furber, and Luis A. Plana. An Investigation into the Security of Self-Timed Circuits. In *ASYNC*, pages 206–215. IEEE Computer Society, May 12-16 2003. Vancouver, BC, Canada. 323

[481] Daheng Yue, Yan Sun, Minxuan Zhang, Shaoqing Li, and Yutong Dai. A Look-Up-Table Based Differential Logic to counteract DPA attacks. In *ASICON*, pages 855–858. IEEE Computer Society, October 20-23 2009. Changsha, Hunan, China. DOI: 10.1109/ASICON.2009.5351561. 323

[482] Ying Zhuang, Nidhal Selmane, Sylvain Guilley, and Jean-Luc Danger. Setup Time Violation Attack on DES and Triple-DES, May 7 2008. 14h00 Session 2B: Fast Abstracts II – Security and Ontologies. Session Chair: Ernesto Jimenez-Merino (http://lsd.ls.fi.upm.es/edcc-7/program). 58