Authentification d'objets à distance

Jean Lancrenon

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 7 Août 2006

Présentée par

Jean LANCRENON

Thèse dirigée par **Roland GILLARD**
et codirigée par **Xavier-François ROBLOT**

préparée au sein de **l'Institut Fourier**
et de **l'école doctorale MSTII**

Authentification d'objets à distance

Thèse soutenue publiquement le 22 juin 2011 ,
devant le jury composé de :

M. Serge VAUDENAY
Professeur au Swiss Federal Institute of Technologies à Lausanne en Suisse
Président

M. Jean-Marc COUVEIGNES
Professeur à l'Université de Toulouse II, Le Mirail
Rapporteur

M. David POINTCHEVAL
Professeur à l'Ecole Normale Supérieure à Paris
Rapporteur

M. Pascal LAFOURCADE
Maître de conférences à l'Université Joseph Fourier à Grenoble
Examinateur

M. Roland GILLARD
Professeur émérite à l'Université Joseph Fourier
Directeur de thèse

M. Xavier-François ROBLOT
Maître de conférences à l'Université Claude Bernard à Lyon
Co-Directeur de thèse

A Rachel

Table des matières

Liste des jeux

Chapitre 1

Chapitre 2

Chapitre 3

Remerciements

Tout d'abord, je tiens beaucoup à remercier Roland Gillard pour avoir accepté de diriger ce travail. Nombreuses ont été les fois où je lui ai promis quelque chose qui finalement soit n'est arrivé qu'avec un certain retard, soit n'est pas arrivé du tout, et j'ai toujours beaucoup apprécié sa patience et sa gentillesse à mon égard. Il m'a également beaucoup soutenu en période de doute et je ne saurais trop lui en être reconnaissant. Je souhaite également remercier Xavier-François Roblot, mon codirecteur de thèse. Les échanges que j'ai eu avec lui et Christophe Delaunay ont toujours été fructueux et son encouragement constant. Mes remerciements vont également à Philippe Elbaz-Vincent qui m'a orienté vers ce projet de thèse ; rien ne serait arrivé sans cela. Merci aussi à Thierry Fournel, dont l'enthousiasme débordant m'a poussé à faire mes premières publications.

Je suis très honoré que David Pointcheval et Jean-Marc Couveignes aient accepté la pénible tâche de rapporter ce travail. Qu'ils en soient profondément remerciés. Merci aussi à Pascal Lafourcade et Serge Vaudenay pour leur présence au sein du jury.

Je voudrais remercier l'Institut Fourier, qui m'a accueilli pendant ces quatre années et m'a permi d'être dans d'excellentes conditions de travail tout le long. Merci aussi à l'ensemble des thésards pour l'ambiance agréable. Le temps passé assis autour de quelques tasses de café (au bureau) ou de plusieurs bières (pas au bureau) n'est jamais du temps perdu.

Je suis extrêmement reconnaissant à la Région Rhônes-Alpes qui a accepté de financer ce projet ainsi que la société Signoptic qui l'a initié. Qu'ils en soient remerciés.

Enfin, je voudrais remercier ma famille : mes parents qui m'ont toujours poussé et encouragé quand il le fallait et surtout (SURTOUT !) ma merveilleuse femme, Rachel, qui m'a toujours encouragé et cru en moi même pendant mes périodes de moral à zéro.

Préambule

Cette thèse est consacrée à la description et à l'étude de la sécurité de divers protocoles destinés à faire de l'authentification d'objets physiques à distance. Nous partons de la possibilité d'extraire d'un objet un flux de données qui lui est propre codé en binaire. Cette extraction est faite une fois avant la mise en circulation de l'objet, et le résultat est stocké dans une base de données sécurisée se trouvant dans un serveur central. Puis, à chaque fois que cela s'avère nécessaire, un lecteur mobile communicant avec le serveur central extrait de l'objet un autre exemplaire de ce flux de données et l'envoie au serveur pour faire une comparaison.

L'objectif des protocoles proposés est de pouvoir réaliser une authentification en garantissant d'une part que les informations envoyées et reçues par le lecteur n'ont pas été manipulées par un adversaire extérieur et d'autre part sans révéler l'identité de l'objet testé à un tel adversaire, ou même, modulo certaines hypothèses raisonnables, aux composantes du système. Nous nous sommes fixés de plus comme objectif d'utiliser des méthodes de cryptographie sur courbe elliptique pour pouvoir profiter des bonnes propriétés de ces dernières, notamment une sécurité accrue par rapport à la taille des clefs utilisées.

Nous présentons plusieurs protocoles atteignant l'objectif et établissons pour presque tous une preuve théorique de leur sécurité. Pour ce faire, il nous a été nécessaire de développer une nouvelle caractérisation d'une notion standard de sécurité. Nous terminons par un tableau comparatif de performances pour les protocoles qui ont été simulés.

Chapitre 1

Introduction

Cette thèse est la présentation de deux protocoles permettant d'authentifier des objets loin d'une autorité compétente, et ce de manière sécurisée. Nous allons commencer par expliquer un peu de quoi il s'agit.

1.1 Qu'est-ce que l'authentification d'objets ?

1.1.1 Généralités et problématique

L'authentification d'un objet est le fait d'établir si cet objet est authentique ou non. Il n'est pas du ressort de ce travail d'étudier le sens du mot "authentique" ; aussi, nous nous bornerons à considérer le cas où pour effectuer une telle authentification, une information supplémentaire est nécessaire. Plus précisément, il faut pouvoir comparer un ou plusieurs traits de l'objet avec d'autres données pour aboutir à une conclusion certaine, ou du moins presque certaine. On peut citer les exemples suivants. Un joaillier détermine l'authenticité d'une pierre précieuse en comparant sa texture à celle d'une pierre de même nature et dont il est sûr de la provenance. L'authenticité d'un billet de banque s'effectue en vérifiant que les structures devant s'y trouver intégrées s'il est valable le sont bel et bien. Une signature peut être authentifiée en la comparant de manière plus ou moins fine avec celle de la personne dont elle prétend provenir. Dans tous ces exemples, une information annexe sert de garantie à l'authenticité de l'objet étudié.

Essayons d'isoler à présent la notion *d'authentification sécurisée.* Quelles propriétés doit-on demander à la procédure de vérification ? Elle doit offrir de très grandes garanties de répondre correctement afin d'offrir le plus de certitude possible. Bien entendu, plus l'objet à authentifier possède de valeur à priori, et plus cette certitude doit être grande. Cette définition informelle implique en particulier la propriété suivante : *la procédure doit s'appuyer sur une donnée suffisamment intrinsèque à l'objet de sorte que toute tentative "fine" de reproduction de l'objet en question fasse inévitablement varier cette donnée.* (L'ad-

jectif "fine" est employé pour exclure des cas de contrefaçon trop grossière. Il faut faire appel au bon sens des gens.) Autrement dit, toute copie, aussi bien exécutée soit-elle, oblige la procédure à répondre par la négative. Reprenons les exemples déjà cités. Les formations naturelles de cristaux sont parfaitement distingables de leurs contreparties artificielles. La technologie utilisée dans les billets de banque est très difficile à reproduire. Enfin, des procédures existent pour distinguer une fausse signature d'une vraie. On peut donc informellement parler d'authentification sécurisée lorsque celle-ci est effectuée via une donnée intrinsèque à l'objet étudié, et difficile à reproduire sans détruire ou dénaturer la copie.

Passons enfin à *l'authentification sécurisée à distance.* Le paragraphe précédent montre que lorsqu'une authentification est souhaitée il faut faire une comparaison entre des données intrinsèques à l'objet et certaines informations extérieures. Il faut donc que ces dernières soient fournies au vérifieur. Il n'est pas raisonnable de penser qu'on puisse avoir ces informations sur soi en permanence ; elles doivent donc être communiquées au moment de l'authentification.

Il y a plusieurs possibilités pour cela. Le vérifieur peut aller les chercher en lieu sûr, ou se les faire communiquer par un autre moyen. Souvent, le premier choix n'est pas pratique donc il faut trouver un moyen de communication. Une première possibilité est d'avoir une copie de ces informations sur l'objet lui même. Il faut alors s'assurer de plus que ces données imprimées sont également authentiques.

Une deuxième possibilité, celle que l'on étudie dans ce travail, est de faire apparaître une entité supplémentaire : une troisième partie (en plus de l'objet et du vérifieur) dont le premier rôle est de stocker les informations annexes utilisées pour pouvoir effectuer une authentification correcte, et le deuxième rôle est d'interagir avec le vérifieur, de manière honnête et sécurisée, afin que ce dernier puisse obtenir les informations comparatives dont il a besoin. Cette méthode exige plus d'infrastructure que celle citée en premier. Cependant, elle s'applique à priori à une plus grande classe d'objets. En effet, on peut l'utiliser pour authentifier des produits sur lesquels il n'est pas possible d'apposer une marque sans faire de dommages. D'autre part, comme elle présuppose qu'une authentification se fait nécessairement au coût d'une communication avec un serveur central, on peut profiter de cette communication pour assurer que le lecteur qui contacte le serveur est lui même affilié de manière licite au système. Ainsi, on allège le niveau de sécurisation de ce dernier qui opère nécessairement en milieu hostile.

Il convient maintenant de préciser ce que l'on entend par une interaction "honnête et sécurisée". On peut supposer que le vérifieur fait confiance à la troisième entité, que l'on appellera le serveur dans la suite. Il lui fait confiance en le sens suivant : il part du principe que le serveur va lui envoyer les informations dont il a besoin, et non de fausses informations. En pratique, un commerçant fait confiance : *i)* aux appareils avec lesquels il scanne les billets de banque et *ii)* à la Banque de France (ou d'ailleurs) qui assure que les billets sont sécurisés convenablement. De même, si un joaillier a besoin d'une photographie

microscopique précise d'un bijou spécifique, il fait confiance aux employés de sa société pour la lui envoyer en cas de besoin.

Cet aspect des choses décrit ce que l'on attend de la part d'un serveur honnête. La sécurisation est plutôt une notion portant sur la transmission même de l'information. Il est clair que si l'on veut être sûr de l'information que l'on reçoit, il faut pouvoir garantir qu'elle n'a pas été manipulée par une entité en cours de transmission. Il faut aussi garantir que la requête faite par le vérifieur soit correctement transmise au serveur. On peut transmettre physiquement une information, auquel cas il faut faire confiance au transporteur de cette information, et on peut la transmettre par voie informatique, auquel cas il faut faire appel à une certaine forme de cryptographie pour garantir, sinon l'arrivée à bon port de l'information, au moins son intégrité. Dans certains cas, par exemple pour tester l'authenticité d'objets de très grande valeur, il peut être également souhaitable d'avoir une procédure garantissant un maximum de confidentialité. Autrement dit, on veut que l'identité présupposée de l'objet ne soit révélée qu'aux entités autorisées. Là aussi, on peut choisir soit de faire confiance à un transporteur, soit de faire appel à la cryptographie. C'est cette dernière voie que nous suivons.

Dans ce travail, la problématique est en fin de compte la suivante. On part de la possibilité d'extraire des informations intrinsèques à un objet pour ainsi avoir une empreinte non modifiable de celui-ci, et on souhaite imaginer un ou plusieurs protocoles cryptographiques qui opèrent en deux temps :

i) dans un premier temps, avant de mettre un produit en libre circulation, on extrait de ce produit les données intrinsèques que l'on stocke dans un serveur central ;

ii) dans un deuxième temps, quand le produit est en circulation - donc "loin" du serveur central - on utilise un appareil léger, maniable et capable de communiquer avec le serveur pour extraire à nouveau du produit les mêmes données et les comparer à celles stockées. Le résultat de la comparaison donne alors la réponse. Enfin, cette procédure doit se faire via des communications vérifiablement authentiques et en préservant le plus possible l'anonymat de l'objet.

1.1.2 Aspect général des protocoles

Il est maintenant nécessaire de mettre en place un peu de terminologie et quelques notations. Pour tout objet $\mathcal{O}$, nous appellerons systématiquement dans la suite un flux de données intrinsèque qui peut en être extrait un *flux morphométrique de* $\mathcal{O}$ ou (abusivement) une morphométrie de $\mathcal{O}$. La raison pour laquelle nous ne pouvons parler de "la" morphométrie de $\mathcal{O}$ sera expliquée plus loin.

D'autre part, nous désignerons par

- $\mathcal{O}$ un objet à authentifier,
- $f_{\mathcal{O}}$ sa morphométrie,
- $\mathcal{R}$ le lecteur de morphométrie,
- $\mathcal{AS}$ le serveur d'authentification, et

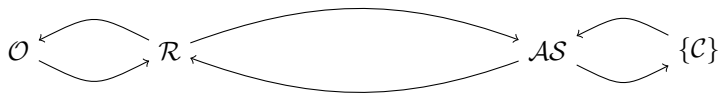


FIGURE 1.1 – Le schéma général

• $\{\mathcal{C}\}$ un ensemble de composantes annexes dont les rôles ne seront complètement décrits que plus loin (en fait, cela dépendra du protocole).

Ces composantes sont réparties en deux milieux : l'objet $\mathcal{O}$ et le lecteur $\mathcal{R}$ opèrent en milieu *hostile* modélisant l'environnement de l'objet une fois que ce dernier a été mis en circulation, et le serveur d'authentification $\mathcal{AS}$ et les $\mathcal{C}$ opèrent en milieu *sécurisé* modélisant l'ensemble des fonctions au sein de l'autorité compétente chargée d'authentifier l'objet. Typiquement, $\mathcal{AS}$ sert de point d'entrée pour l'ensemble des communications et gère les demandes d'authentification de plusieurs lecteurs affiliés au système, et $\{\mathcal{C}\}$ contient la base de données constituée des flux morphométriques de référence ainsi que d'autres entités jouant des rôles techniques dans la procédure d'authentification. Nous supposerons que $\mathcal{R}$ communique avec $\mathcal{AS}$ et que $\mathcal{AS}$ communique avec les $\mathcal{C}$; la figure 1.1 montre le schéma général jusque-là, les flèches représentant des communications.

Nous pouvons déjà esquisser ici un protocole d'authentification morphométrique à distance. Nous nous préoccuperons de sa sécurité ensuite. Tous nos protocoles se dérouleront ainsi.

Protocole générique (fig. 1.1) :

Après sa fabrication, $\mathcal{O}$ est enregistré au sein du système qui stocke dans $\mathcal{C}$ une copie de $f_{\mathcal{O}}$ à un indice $i_{\mathcal{O}}$. Cet indice devient alors l'identité système de $\mathcal{O}$, et est imprimé dessus. Ensuite, l'objet est mis en circulation. Puis, si une personne munie d'un lecteur $\mathcal{R}$ conforme souhaite vérifier l'authenticité de $\mathcal{O}$, elle extrait une morphométrie f' de $\mathcal{O}$, lit l'indice $i_{\mathcal{O}}$, et envoie le couple $(f', i_{\mathcal{O}})$ à $\mathcal{AS}$ qui le transmet aux $\mathcal{C}$. Ces derniers utilisent $i_{\mathcal{O}}$ pour déterminer à quelle référence stockée doit être comparée f'. Il ne leur reste plus qu'à effectuer la comparaison, et renvoyer la réponse à $\mathcal{AS}$ qui la transmet à $\mathcal{R}$.

1.2 Morphométrie, biométrie et modèle de sécurité

Il s'agit maintenant de décrire plus en détail nos exigences sécuritaires. Nous avons évoqué plus haut l'authenticité vérifiable des communications et l'anonymat de l'objet ; pour voir comment intégrer ces exigences dans le schéma ci-dessus, nous allons commencer par décrire un problème auquel le notre res-

semble beaucoup et qui nous a d'ailleurs servi de modèle pour notre travail : l'authentification d'individus à l'aide de données biométriques.

Le principe de fonctionnement est exactement le même ; on remplace simplement l'objet par une personne et la morphométrie par la biométrie. Sinon, le schéma de la figure 1.1 reste inchangé. En fait, les résultats de cette thèse ont pour point de départ des protocoles d'authentification biométrique imaginé par Bringer *et al.* dans [5] et [6], protocoles dont nous nous inspirons largement. A partir du moment où le principe est identique il est naturel de chercher simplement à adapter le modèle de sécurité au cas morphométrique.

Nous allons commencer par rappeler en détail les schéma de fonctionnement et modèle de sécurité qui se trouvent dans [5]. Nous ferons ensuite une petite comparaison entre les données biométriques et morphométriques. Enfin, nous expliquerons comment ces différences dictent nos choix dans notre modèle de sécurité, que nous présentons à la fin du chapitre en termes de jeux d'attaque.

1.2.1 Un schéma d'authentification biométrique

Ce qui suit résume une partie des résultats dans [5] et [6].

Infrastructure

Reprenons des notations similaires à celles de 1.1.2. En particulier, une donnée biométrique extraite d'un individu sera appelée donnée ou flux biométrique de ce même individu.

• Nous noterons $\mathcal{U}$ un individu à authentifier enregistré au sein du système. En particulier, il possède une identité-système $SID_{\mathcal{U}}$ et une biométrie de référence $b_{\mathcal{U}}$ provenant de $\mathcal{U}$ est stockée dans la base de données. Précisément, il existe un indice $i_{\mathcal{U}}$ tel que $\mathcal{AS}$ possède $(SID_{\mathcal{U}}, i_{\mathcal{U}})$ et la base de données possède $(i_{\mathcal{U}}, b_{\mathcal{U}})$.

• Nous désignerons par $\mathcal{R}$ le lecteur de biométrie. Son rôle est d'extraire une donnée biométrique temporaire - notée b' dans la suite - de quiconque se présente à lui pour faire une demande d'authentification. Il prend comme donnée supplémentaire notamment l'identité sytème de l'utilisateur, et communique avec le serveur d'authentification.

• Le serveur d'authentification sera noté $\mathcal{AS}$. Il gère les demandes d'authentification provenant de plusieurs lecteurs et sert de point d'entrée à toutes les communications entre le milieu hostile contenant l'utilisateur et le lecteur et le milieu serveur constitué de $\mathcal{AS}$, $\mathcal{DB}$ et $\mathcal{M}$. Enfin, il contient une liste de la forme $(SID_{\mathcal{V}}, i_{\mathcal{V}})_{\mathcal{V}}$ où $\mathcal{V}$ parcourt l'ensemble des utilisateurs.

• La base de données biométrique sera appelée $\mathcal{DB}$. Elle communique avec $\mathcal{AS}$ et possède la liste $(i_{\mathcal{V}}, b_{\mathcal{V}})_{\mathcal{V}}$.

• Enfin, $\mathcal{M}$ est une composante appelée comparateur. Lors d'une authentification, elle prend en entrée des données dépendant de $b_{\mathcal{U}}$ et b' lui permettant de comparer ces deux flux. C'est elle qui détermine le résultat de l'authentification qu'elle retourne à $\mathcal{AS}$.

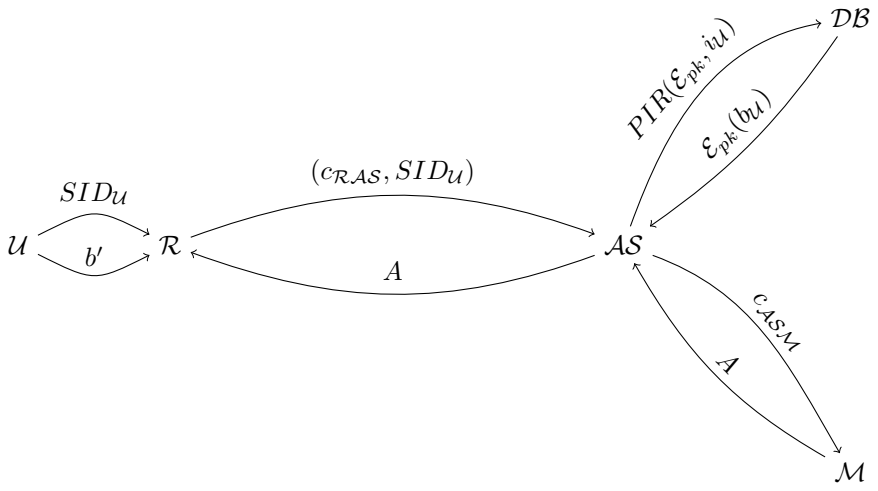


FIGURE 1.2 – Le schéma du protocole dans [5]

Exigences sécuritaires

Le principal souci des auteurs dans [5] est de pouvoir effectuer une authentification biométrique tout en protégeant l'anonymat de l'utilisateur. Ils partent du principe que les données biométriques d'un individu doivent être considérées comme des données publiques car il n'est en pratique pas difficile de les obtenir. Nous reviendrons sur ces questions un peu plus loin. L'hypothèse formelle qui est faite est donc que toute entité malicieuse cherchant à déterminer qui essaye de s'authentifier à un moment donné possède une liste de la forme $(RID_{\mathcal{V}}, b^1_{\mathcal{V}})_{\mathcal{V}}$ où $\mathcal{V}$ parcourt la liste des utilisateurs du système, $b^1_{\mathcal{V}}$ est une autre donnée biométrique correspondant à $\mathcal{V}$, et $RID_{\mathcal{V}}$ est l'identité réelle de $\mathcal{V}$. Ainsi le problème se réduit à **protéger les liens entre $RID_{\mathcal{U}}$, $b_{\mathcal{U}}$ et b' lors d'une ronde d'authentification.** Ce problème est résolu justement grâce à l'introduction de l'identité système $SID_{\mathcal{U}}$ et du comparateur $\mathcal{M}$ en conjonction avec des hypothèses faites sur les interactions possibles entre $\mathcal{AS}$, $\mathcal{DB}$ et $\mathcal{M}$.

Il convient maintenant d'expliquer quelles sont ces hypothèses et contre quels adversaires on veut se prémunir.

Les deux hypothèses qui sont faites concernent le côté serveur du schéma. Elles traduisent en partie le fait que ce dernier soit raisonnablement sécurisé à priori.

Hypothèse 1 *Il est supposé que les composantes $\mathcal{AS}$, $\mathcal{DB}$ et $\mathcal{M}$ ne dévient pas du protocole lors d'une tentative d'authentification. Autrement dit, elles ne font qu'une observation passive des communications.*

Hypothèse 2 *Il est supposé que les composantes $\mathcal{AS}$, $\mathcal{DB}$ et $\mathcal{M}$ ne partagent à aucun moment leurs informations.*

Passons aux exigences sécuritaires. Elles ne concernent que la confidentialité de l'authentification et incorporent comme adversaires possibles les composantes mêmes du protocole d'authentification. La suite de la thèse contenant des définitions techniques précises de notions analogues à celles qui suivent, nous nous bornons ici à en fournir des versions informelles.

Exigence 1 *Lors d'une tentative d'authentification, aucun adversaire $\mathcal{A}$ extérieur au système ne peut obtenir d'information sur l'identité réelle de la personne qui cherche à s'authentifier.*

Exigence 2 *Lors d'une tentative d'authentification, aucune des trois composantes $\mathcal{AS}$, $\mathcal{DB}$ et $\mathcal{M}$ ne peut obtenir d'information sur l'identité réelle de la personne qui cherche à s'authentifier.*

Le protocole

Avant de décrire le protocole, nous devons mettre en place quelques notations supplémentaires. Nous désignerons par

• $\mathcal{E}_{pk}(.)$ un algorithme de chiffrement asymétrique, sémantiquement sûr (voir le chapitre 3), chiffrant bit-par-bit et homomorphique (i.e. pour deux vecteurs binaires m et m' de même longueur on a $\mathcal{E}_{pk}(m) \odot \mathcal{E}_{pk}(m') = \mathcal{E}_{pk}(m \oplus m')$, où $\odot$ désigne une loi de groupe dont l'espace des chiffrés est muni et $\oplus$ est le XOR habituel) et (pk, sk) une paire (*clef publique*, *clef privée*) pour cet algorithme, et

• $PIR(\mathcal{E}_{pk}, .)$ un protocole de récupération confidentielle d'information (*private information retrieval*) dérivé de $\mathcal{E}_{pk}(.)$, i.e. une procédure par laquelle on peut récupérer le chiffrement d'un objet stocké dans une base de données sans que celle-ci sache lequel.

Nous supposerons que l'information biométrique extraite est convertie en vecteur binaire, i.e. b' et $b_{\mathcal{U}}$ sont dans $\{0,1\}^*$, et ont même longueur. Enfin, la comparaison entre vecteurs est effectuée en déterminant le poids de Hamming du XOR (noté $\oplus$) de ces deux vecteurs et en comparant ce poids à une limite τ : deux vecteurs correspondent si et seulement si ce poids n'excède pas τ.

Enfin, nous supposerons que le lecteur possède $\mathcal{E}(.)$ et pk, que $\mathcal{AS}$ et $\mathcal{DB}$ sont équipés pour interagir selon l'algorithme $PIR(\mathcal{E}_{pk}, .)$, et que $\mathcal{M}$ possède sk.

Le protocole se déroule alors comme suit (voir la figure 1.2) :

• L'utilisateur $\mathcal{U}$ s'identifie auprès du lecteur $\mathcal{R}$ par $SID_{\mathcal{U}}$ et $\mathcal{R}$ extrait b'.

• Le lecteur calcule $c_{\mathcal{RAS}} := \mathcal{E}_{pk}(b')$ et envoie $(c_{\mathcal{RAS}}, SID_{\mathcal{U}})$ à $\mathcal{AS}$.

• Le serveur $\mathcal{AS}$ obtient $i_{\mathcal{U}}$ de $SID_{\mathcal{U}}$ et interagit avec la base de données $\mathcal{DB}$ selon les spécifications de $PIR(\mathcal{E}_{pk}, i_{\mathcal{U}})$. Le résultat de cette interaction est que $\mathcal{AS}$ reçoit $\mathcal{E}_{pk}(b_{\mathcal{U}})$ sans avoir donné $i_{\mathcal{U}}$ en clair à $\mathcal{DB}$.

• Le serveur calcule $\mathcal{E}_{pk}(b' \oplus b_{\mathcal{U}}) = \mathcal{E}_{pk}(b') \odot \mathcal{E}_{pk}(b_{\mathcal{U}})$, choisit une permutation σ uniformément au hasard, et calcule $c_{\mathcal{ASM}} := \sigma\big(\mathcal{E}_{pk}(b' \oplus b_{\mathcal{U}})\big)$ (où la permutation

est appliquée chiffré par chiffré, de sorte que lorsque chaque composante est individuellement déchiffrée, le résultat est $\sigma(b' \oplus b_{\mathcal{U}}))$ qu'il envoie au comparateur $\mathcal{M}$.

• Le comparateur utilise sk pour calculer $\sigma(b' \oplus b_{\mathcal{U}})$ et détermine le poids de Hamming du vecteur obtenu. Il envoie sa réponse A à $\mathcal{AS}$ qui la transmet à $\mathcal{R}$.

Sécurité

Exposons maintenant pourquoi les exigences sécuritaires sont satisfaites grâce aux hypothèses et à l'infrastructure. Supposons qu'un utilisateur $\mathcal{U}$ enregistré au sein du système fasse une demande d'authentification, et examinons les possibilités des attaquants considérés par le modèle.

Un adversaire extérieur $\mathcal{A}$

Par définition, $\mathcal{A}$ possède, nous l'avons vu plus haut, la liste $(RID_{\mathcal{V}}, b^1_{\mathcal{V}})_{\mathcal{V}}$ relative aux utilisateurs ($RID_{\mathcal{V}}$ et $b^1_{\mathcal{V}}$ désignant respectivement l'identité réelle de $\mathcal{V}$ et une biométrie correpondant à $\mathcal{V}$) et son but est de déterminer $RID_{\mathcal{U}}$. Toujours par définition, $\mathcal{A}$ observe les canaux de communication entre les diverses composantes. Il possède donc les informations suivantes provenant des communications :

- $\big(\mathcal{E}_{pk}(b'), SID_{\mathcal{U}}\big)$ provenant du canal entre $\mathcal{R}$ et $\mathcal{AS}$,
- $\big(PIR(\mathcal{E}_{pk}, i_{\mathcal{U}}), \mathcal{E}_{pk}(b_{\mathcal{U}})\big)$ venant de $\mathcal{AS}$ et $\mathcal{DB}$ et
- $\Big(\sigma\big(\mathcal{E}_{pk}(b' \oplus b_{\mathcal{U}})\big), A\Big)$ de $\mathcal{AS}$ et $\mathcal{M}$.

Que peut faire $\mathcal{A}$ de ces données ? Pour obtenir $RID_{\mathcal{U}}$, il peut essayer de comparer les $b^1_{\mathcal{V}}$ à $b_{\mathcal{U}}$ et b', mais ces deux derniers sont toujours chiffrés. Ensuite, il ne peut rien obtenir de $SID_{\mathcal{U}}$ car il n'y a à priori aucun lien entre $RID_{\mathcal{U}}$ et $SID_{\mathcal{U}}$. La donnée $PIR(\mathcal{E}_{pk}, i_{\mathcal{U}})$ ne révèle au plus que $i_{\mathcal{U}}$ que $\mathcal{A}$ ne peut exploiter puisqu'il ne possède pas $(i_{\mathcal{V}}, b_{\mathcal{V}})_{\mathcal{V}}$. Enfin, A ne contient aucune information relative à l'identité. L'exigence 1 est donc vérifiée. □

Une composante serveur malicieuse

La première remarque à faire est que par l'hypothèse 2, $\mathcal{AS}$ et $\mathcal{DB}$ ne partagent pas les listes qu'elles possèdent, à savoir respectivement $(SID_{\mathcal{V}}, i_{\mathcal{V}})_{\mathcal{V}}$ et $(i_{\mathcal{V}}, b_{\mathcal{V}})_{\mathcal{V}}$. De même, $\mathcal{M}$ garde la clef de déchiffrement sk pour lui-même. L'hypothèse 2 élimine donc des cas triviaux où la sécurité n'est plus assurée. Il faut maintenant regarder ce qui se passe composante par composante.

Le serveur d'authentification $\mathcal{AS}$ reçoit au total $\big(\mathcal{E}_{pk}(b'), SID_{\mathcal{U}}, \mathcal{E}_{pk}(b_{\mathcal{U}}), A\big)$. Nous venons de remarquer qu'il ne peut pas déchiffrer, donc les chiffrés ne lui apportent rien et de manière analogue au cas d'un adversaire extérieur, $SID_{\mathcal{U}}$ et A non plus.

La base de données reçoit des informations de $\mathcal{AS}$ lors de l'interaction spécifiée par $PIR(\mathcal{E}_{pk}, i_{\mathcal{U}})$. Le PIR étant par définition un algorithme permettant ici d'extraire $\mathcal{E}_{pk}(b_{\mathcal{U}})$ de $\mathcal{DB}$ sans révéler $i_{\mathcal{U}}$ à $\mathcal{DB}$, cette dernière ne reçoit rien.

Enfin, le comparateur $\mathcal{M}$ peut déchiffrer toutes les communications mais la seule qu'il reçoit lors du protocole est $\sigma\big(\mathcal{E}_{pk}(b' \oplus b_{\mathcal{U}})\big)$, et il ne peut en demander d'autre en vertu de l'hypothèse 2. Il ne voit en fin de compte que la permutation aléatoire du XOR de deux vecteurs binaires.

Il résulte des trois explications précédentes que l'exigence 2 est également vérifiée. □

1.2.2 Morphométrie vs biométrie

Nous allons maintenant examiner plus en détail les similarités et différences entre les données morphométriques et biométriques, et comment ces données peuvent être manipulées par des primitives cryptographiques. Précisément, nous allons regarder quels problèmes cryptographiques sont posés par la biométrie, voir s'ils se transportent au cas morphométrique et, si tel est le cas, de quelle manière. Nous en déduirons une adaptation du modèle vu informellement ci-dessus au cas qui nous intéresse.

Les problèmes cryptographiques posés par la biométrie sont de deux types essentiellement. Il y a un problème de *précision dans la mesure des données,* et un problème de *sécurisation d'une information sensible.*

Mesures

Le problème posé par la mesure d'une donnée biométrique est simplement le fait que lorsque deux telles mesures sont effectuées, elles ne sont que rarement exactement les mêmes. Ceci est dû à un grand nombre de choses : la continuité du milieu mesuré, le fait que ce soit une donnée biologique donc variant au cours du temps, les erreurs de lecture etc.... Le mieux que l'on puisse espérer à l'heure actuelle est que deux données biométriques convenablement extraites d'un même individu soient "assez proches" pour conclure presque certainement qu'il s'agit bien de la même personne. Or, une variabilité même minime dans la mesure n'est pas compatible à beaucoup de primitives cryptographiques. On peut prendre l'exemple classique d'une fonction de hachage H servant à protéger un mot de passe : pour accéder à un service, un utilisateur $\mathcal{U}$ entre son mot de passe $pw_{\mathcal{U}}$ à l'écran et le serveur gérant l'accès au service reçoit non pas $pw_{\mathcal{U}}$ mais $h_{\mathcal{U}} := H(pw_{\mathcal{U}})$. La fonction H étant cryptographique elle a (entre autres) la propriété qu'une petite perturbation de $pw_{\mathcal{U}}$ change massivement $h_{\mathcal{U}}$. On conclut que pour utiliser la biométrie en tant que mot de passe, il faut soit trouver un moyen de l'extraire parfaitement soit faire en sorte que le système tolère une (petite) erreur.

Ces considérations doivent être prises en compte aussi au niveau morphométrique. C'est toujours une question de continuité et de variation physique du milieu mesuré. Ce fait justifie que nous ne pouvons parler de "la" morphométrie d'un objet (ou d'ailleurs de "la" biométrie d'un individu).

Informations sensibles

L'information sensible dont il est question est la biométrie elle-même car la biométrie d'une personne la détermine uniquement. Son utilisation pose donc des problèmes éthiques considérables étant donnée la nature de ce qui est mesuré, et il est nécessaire d'intégrer à toute primitive s'appuyant sur elle des mécanismes pour la protéger. Un paradoxe apparaît alors dans cette situation : bien que la biométrie d'un individu soit une donnée excessivement sensible, il n'est pas raisonnable de la considérer en pratique comme une donnée secrète. En effet, elle n'est pas si difficile que ça à "voler" : chacun laisse des empreintes digitales partout par exemple. Un protocole d'authentification biométrique a donc aussi la tâche - un peu contre-nature en cryptographie - de traiter une donnée publique comme une donnée secrète.

En ce qui concerne la morphométrie, ce problème ne se pose pas car la morphométrie d'un objet inanimé n'est pas une valeur sensible bien qu'elle caractérise l'objet. Par ailleurs, on la considère toujours comme publique vu que les méthodes et algorithmes servant à l'extraire le sont : une fois l'objet en circulation, toute personne l'ayant en main peut potentiellement en extraire une morphométrie.

Adaptation du modèle de sécurité

Nous avons vu que le protocole décrit dans [5] permet de réaliser l'authentification d'un individu en préservant son anonymat. Nous avons aussi vu au paragraphe 1.1.1 qu'il est souhaitable de pouvoir faire de même lors de l'authentification d'un objet. Maintenant, cette exigence a été satisfaite dans le protocole biométrique en combinant le chiffrement des flux biométriques concernés $b_{\mathcal{U}}$ et b' avec l'utilisation d'une identité système $SID_{\mathcal{U}}$. Cette dernière doit être tenue par l'utilisateur puisqu'il s'identifie auprès du serveur avec, et il est clair que si $\mathcal{U}$ permet la mise en circulation de $SID_{\mathcal{U}}$ de sorte à pouvoir remonter à son identité réelle, la sécurité n'est plus assurée. Il en résulte que pour que le protocole biométrique exposé ci-dessus préserve l'anonymat, **il est impératif que $\mathcal{U}$ garde secrète son identité système.** C'est là la première des deux différences majeures auxquelles nous sommes confrontées ici : un objet ne peut pas possèder et protéger lui-même une identité système. **L'identité système d'un objet coïncide donc avec son identité réelle, et nous devons trouver un moyen de la protéger dès la première communication entre le lecteur et le serveur.**

La deuxième différence majeure réside dans la protection de la biométrie en vertu du fait que ce soit une information sensible. Dans [5] les auteurs font la remarque que la sensibilité des données biométriques est suffisante pour justifier la recherche d'autres protocoles permettant de réaliser les mêmes objectifs tout en stockant dans la base de données des flux biométriques *déjà chiffrés*. L'article [6] contient d'ailleurs un tel protocole. Vu que la sensibilité des données morphométriques est minimale, leur protection absolue ne nous concerne pas. **Nous allons même exploiter ce fait en permettant quelques uns de nos**

protocoles d'opérer à l'aide d'une infrastructure contenant plusieurs copies de la base de données morphométriques, les flux étant stockés en texte clair.

1.3 Objectifs et modèle formel de sécurité

Maintenant que nous avons vu un exemple de protocole d'authentification biométrique et que nous avons analysé (un peu) les différences inhérentes entre les méthodes biométriques et morphométriques, détaillons les objectifs que nous nous proposons d'atteindre dans un protocole d'authentification d'objet.

Confidentialité

Comme pour un protocole d'authentification biométrique, nous voulons garantir que certains adversaires, aussi bien intérieur qu'extérieur au système, **ne peuvent pas déterminer quel objet cherche à s'authentifier.** Cela peut être une exigence pour des objets de grande valeur ou tenu au secret. Le protocole biométrique ci-dessus doit être modifié dans ce cas pour tenir compte du fait qu'un objet n'a pas d'identité système qu'il puisse garder cachée lorsqu'il est en circulation.

Dans le cadre de la confidentialité, nous supposons que les adversaires extérieurs sont des adversaires faisant juste des observations passives des communications entre les différentes composantes lors d'une ronde d'authentification.

Intégrité de la réponse finale

Nous voulons de plus garantir que la réponse finale de la ronde d'authentification soit **intègre**, c'est-à-dire qu'il n'est pas possible pour un adversaire extérieur de modifier la réponse finale sans que l'une au moins des composantes du système ne s'en aperçoive. On remarque que cet objectif n'est pas du tout adressé dans le protocole ci-dessus.

Dans le cadre de l'intégrité de la réponse, nous supposons que les adversaires extérieurs sont des adversaires actifs observant et modifiant éventuellement les communications entre $\mathcal{R}$ et $\mathcal{AS}$ seulement.

Donnons ici une fois pour toutes une description complète des composantes qui se trouvent dans les protocoles présentés dans la suite, puis d'un modèle formel de sécurité. Cette section contient l'essentiel des notations.

1.3.1 Les acteurs des protocoles d'authentification

Les protocoles construits dans ce travail suivent tous le schéma de la figure 1.1 mais diffèrent au niveau des composantes notées $\mathcal{C}$. Le premier paragraphe contient une description des composantes $\mathcal{O}$, $\mathcal{R}$ et $\mathcal{AS}$ présentes dans tous les protocoles et le deuxième examine les $\mathcal{C}$ selon le protocole envisagé. Quand aux méthodes cryptographiques utilisées par chaque composante, elles dépendent entièrement du protocole et ne seront données qu'au moment de sa présentation.

Composantes communes à tous les protocoles

L'objet physique

Il sera systématiquement désigné par $\mathcal{O}$ (ou $\mathcal{O}'$ lorsqu'il sera dans un contexte où il s'identifie comme étant $\mathcal{O}$, mais doit encore le prouver). Nous ne faisons aucune hypothèse de forme sur cet objet. Il est supposé enregistré au sein du système ; il lui est donc attribué une identité système simplement sous la forme d'un entier naturel noté $i_{\mathcal{O}}$ et imprimé sur lui. Cet entier est lisible par toute entité ayant l'objet en main. Il est supposé aussi qu'une lecture morphométrique de référence a été faite et que le flux $f_{\mathcal{O}}$ qui en est issu est stocké dans une base de données à l'indice $i_{\mathcal{O}}$. Pour plus de clarté dans les notations lors de la présentation des protocoles, on notera parfois ce flux $f_{i_{\mathcal{O}}}$.

Les flux morphométriques

Dans toute la suite de ce mémoire nous considérons que les flux morphométriques sont, après extraction, convertis en vecteurs binaires de longueur $k \in \mathbb{N} - \{0\}$. Tous les flux stockés dans la base de données sont de cette même longueur. Le flux de référence d'un objet $\mathcal{O}$ sera noté $f_{\mathcal{O}}$ ou $f_{i_{\mathcal{O}}}$ selon le contexte.

Les flux extraits lors d'une demande d'authentification seront notés f'. Comme nous l'avons dit précédemment, il est supposé que lorsque deux flux sont extraits d'un même objet, ils ne sont presque jamais égaux. Néanmoins, ils sont "assez proches" au sens suivant : en notant hw le poids de Hamming sur $\{0,1\}^k$, il existe un entier naturel $\tau \leq k$ appelé seuil d'authenticité tel que si $f^1_{\mathcal{O}}$ et $f^2_{\mathcal{O}}$ sont extraits correctement du même objet, on ait $hw(f^1_{\mathcal{O}} \oplus f^2_{\mathcal{O}}) \leq \tau$.

Nous supposons également que si $\mathcal{O}^1$ et $\mathcal{O}^2$ sont deux objets physiquement différents (i.e. même si l'un est une copie industrielle de l'autre) et que $f_{\mathcal{O}^1}$ et $f_{\mathcal{O}^2}$ sont correctement extraits respectivement de $\mathcal{O}^1$ et $\mathcal{O}^2$, les flux vérifient $hw(f_{\mathcal{O}^1} \oplus f_{\mathcal{O}^2}) > \tau$.

Nous supposons enfin que toute entité ayant l'objet en main est capable d'en extraire correctement un flux morphométrique. D'après ce que nous avons déjà vu, en réalité un tel flux ne sera la plupart du temps jamais égal au flux de référence stocké. Cependant, en pratique il obéit aux mêmes propriétés que ce dernier vis-à-vis du seuil τ. Nous pouvons donc sans incovénient considérer que pour tout objet $\mathcal{O}$ enregistré au sein du système, la donnée $(i_{\mathcal{O}}, f_{\mathcal{O}})$ est publique.

Le lecteur

Il est noté $\mathcal{R}$ dans la suite. Il est équipé pour faire deux types de lectures : l'une d'elles permet une obtention **parfaite** de l'indice $i_{\mathcal{O}}$ se trouvant imprimé sur l'objet, et l'autre permet une extraction - la plupart du temps imparfaite - et une conversion en binaire, d'un flux morphométrique correspondant à l'objet.

En pratique il faut considérer la possibilité que $\mathcal{R}$ fasse des erreurs de lecture car cela peut conduire à des résultats d'authentification erronés. C'est une question d'implémentation qui ne concerne pas le fonctionnement de la procédure d'authentification proprement dite donc nous n'en tiendrons pas compte (notamment dans l'étude théorique de la sécurité offerte).

Nous faisons quelques hypothèses sur le lecteur. Toutes les entités du système considèrent qu'il extrait toujours correctement toutes les informations d'un objet donné, et n'en envoie pas de fausse. Cette hypothèse est raisonnable en pratique, où on imagine par exemple qu'un individu voulant faire l'authentification d'un objet est physiquement présent pour vérifier que le lecteur fonctionne correctement.

Le serveur d'authentification

Cette composante, notée $\mathcal{AS}$, est l'entité principale gérant l'ensemble du système. Il sert de point d'entrée à toutes les communications lors d'une authentification. Typiquement, il communique avec plusieurs lecteurs dont il est capable de déterminer s'ils sont affiliés au système ou non. Cette question n'intervenant pas dans le cadre de notre modèle de sécurité, nous ne reviendrons pas dessus. Cependant, le fait que les lecteurs communiquent obligatoirement avec le serveur central lors d'une authentification ouvre la possibilité de profiter de ces communications pour implémenter des protocoles de certification de lecteur.

Nous faisons les hypothèses suivantes sur le serveur d'authentification. Les entités du système considèrent qu'il sait distinguer les lecteurs affiliés au système de ceux qui ne le sont pas. Ils considèrent aussi que lors d'une authentification, il agit comme spécifié par le protocole. Cependant il est capable de conserver en mémoire d'éventuelles communications passées.

Le serveur $\mathcal{AS}$ peut avoir un comportement malicieux, mais limité puisqu'il opère en milieu raisonnablement sécurisé. Nous supposons qu'en observant les communications, il peut essayer de déterminer quel objet essaye de s'authentifier *à priori*.

Composantes spécifiques à certains protocoles

Les bases de données

Ce sont des composantes notées $\mathcal{DB}$ qui contiennent essentiellement les flux de référence extraits. Formellement, une base de données est donc de la forme $(i_{\mathcal{O}}, f_{\mathcal{O}})_{\mathcal{O}}$ où $\mathcal{O}$ parcourt l'ensemble des objets enregistrés. Nous avons vu que chaque objet possède une identité système $i_{\mathcal{O}}$ sous la forme d'un entier, et cet entier n'est autre que l'indice d'appartenance de $f_{\mathcal{O}}$ dans $\mathcal{DB}$. Nous noterons N le nombre total d'objets enregistrés dans une base de données, donc les $i_{\mathcal{O}}$ parcourent $\{1, ..., N\}$.

Le nombre de bases de données varie selon le protocole mais tous les protocoles en possèdent au moins une.

Les bases de données ne communiquent qu'avec le serveur d'authentification. Elles font partie du milieu sécurisé donc comme pour le serveur nous supposerons qu'elles opèrent comme spécifié par le protocole à tout moment. Enfin, nous considérons que leur contenu est entièrement publique : toute entité interne ou externe au système possède la liste $(i_{\mathcal{O}}, f_{\mathcal{O}})_{\mathcal{O}}$ (en pratique il peut s'agir d'une liste équivalente au sens où les indices sont les mêmes, et les flux extraits correpondent sans être toutefois ceux de référence).

Le comportement malicieux que peut avoir une base de données est réduit à essayer de déterminer quel objet à priori cherche à s'authentifier.

Le comparateur

C'est une composante notée $\mathcal{M}$ utilisée dans l'un des protocoles pour effectuer la comparaison entre deux flux morphométriques.

Comme pour les autres composantes du milieu sécurisé, nous supposons que $\mathcal{M}$ ne dévie pas du protocole spécifié : il est supposé qu'il effectue correctement la comparaison entre flux, et donne la réponse appropriée.

De même, comme les autres composantes du système, le comparateur ne peut qu'observer les communications qu'il reçoit, les garder en mémoire, et essayer de déterminer quel objet s'authentifie à un moment donné.

Hypothèses sur les composantes

Il nous faut d'abord décrire en détail les hypothèses faites sur les interactions entre les composantes.

Milieu non sécurisé

Ce qu'on appelle le milieu non sécurisé est l'environnement où se trouvent le lecteur et l'objet. Les adversaires se trouvant dans ce milieu sont des observateurs actifs des communications entre $\mathcal{R}$ et $\mathcal{AS}$. Nous ne supposons pas qu'ils attaquent entre $\mathcal{O}$ et $\mathcal{R}$. Qu'il soient actifs signifie qu'ils observent les communications et peuvent les modifer.

Milieu sécurisé

Le milieu dit sécurisé est celui des communications entre les composantes $\mathcal{AS}$ et les $\mathcal{C}$. Il est sécurisé au sens où il vérifie des hypothèses analogues à celles définies dans le paragraphe 1.2.1 :

Hypothèse 3 *Il est supposé qu'aucune des composantes $\mathcal{AS}$ et $\mathcal{C}$ ne dévie du protocole lors d'une tentative d'authentification. Autrement dit, elles ne font qu'une observation passive des communications.*

Hypothèse 4 *Il est supposé que les composantes $\mathcal{AS}$ et $\mathcal{C}$ ne partagent pas leurs informations entre elles.*

Hypothèse 5 *Il est supposé que toutes les communications entre les composantes $\mathcal{AS}$ et $\mathcal{C}$ sont vérifiablement authentiques.*

1.3.2 Modèle formel de sécurité

La formalisation du modèle de sécurité passe obligatoirement par l'utilisation d'algorithmes. Nous nous plaçons du point de vue de la sécurité fondée sur la théorie de la complexité. Le principal fondement de cette théorie peut se décrire

informellement ainsi : **toute procédure de calcul efficace peut être implémentée en temps polynomial par une machine de Turing probabiliste dans le modèle de calcul uniforme.** C'est la thèse de Church-Turing.

Nous ne reviendrons pas sur la notion de machine de Turing probabiliste opérant en temps polynomial, que nous appellerons dans la suite simplement PPTA (pour *probabilistic polynomial-time algorithm*). Parfois, nous aurons affaire à des machines de Turing déterministes opérant en temps polynomial. Elles seront abbréviées en DPTA (*deterministic polynomial-time algorithm*). Tous les algorithmes dans la suite opèrent en temps polynomial. Un tel algorithme, sans préciser s'il est déterministe ou probabiliste, sera simplement abbrévié en PTA. Une introduction générale à la théorie de la complexité appliquée à la cryptographie peut-être trouvée dans [33] ou [37]. Pour des définitions en termes de théorie de la complexité de beaucoup des primitives de construction de base de la cryptographie, le lecteur pourra consulter [15] et [16].

Conformément au modèle de calcul dans lequel nous nous plaçons, tous les algorithmes que nous considérons, en particulier tous les adversaires contre lequels nous souhaitons nous prémunir, sont des PPTA. Pour la même raison, toutes les composantes du système sont représentés en PTA. Nous pouvons donc former les définitions suivantes.

Définition 1 *Soit $\mathcal{DBG}$ un PTA. On dit que $\mathcal{DBG}$ est un* ***générateur de bases de données*** *si pour tout $n \in \mathbb{N}$, $\mathcal{DBG}$ prend 1^n en entrée et retourne*

$$\big(k, N, \tau, (i, f_i)_i\big)$$

où
i) *k et N et τ sont des entiers, $k \geq 1$, $N \geq 2$, $\tau \in \{0, ..., k\}$ et*
ii) *pour tout $i \in \{1, ..., N\}$, $f_i \in \{0,1\}^k$.*

Si $\mathcal{A}$ et $\mathcal{B}$ désignent des PTA, la notation

$$(m_{\mathcal{A}}, m_{\mathcal{B}}) \leftarrow \big\langle \mathcal{A}(x_{\mathcal{A}}), \mathcal{B}(x_{\mathcal{B}}) \big\rangle$$

désigne l'ensemble des messages retournés par $\mathcal{A}$ et $\mathcal{B}$ lors d'une interaction, où $\mathcal{A}$ et $\mathcal{B}$ prennent comme entrées initiales respectives $x_{\mathcal{A}}$ et $x_{\mathcal{B}}$.

Définition 2 *Un* ***protocole d'authentification morphométrique*** *$\mathcal{PAM}$ est la donnée de*

$$\big(\mathcal{R}_1, \mathcal{R}_2, \mathcal{AS}, (\mathcal{C})_{\mathcal{C}}, \textsc{Infra}, \textsc{Enr}\big)$$

où $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{AS}$, les $\mathcal{C}$, Infra *et* Enr *sont des PTA fonctionnant de la manière suivante. Soit $n \in \mathbb{N}$.*
1) *Pour tout générateur de base de données $\mathcal{DBG}$ et tout $\big(k, N, \tau, (i, f_i)_i\big) \in \mathcal{DBG}(1^n)$ la fonctionnalité* Infra *prend en entrée 1^n, interagit avec $\mathcal{R} := (\mathcal{R}_1, \mathcal{R}_2)$, $\mathcal{AS}$ et les $\mathcal{C}$ et retourne un quadruplet*

$$\big(I_{ext}, I_{\mathcal{R}}, I_{\mathcal{AS}}, (I_{\mathcal{C}})_{\mathcal{C}}\big)$$

2) *La fonctionnalité* ENR *prend en entrée* $\big(k, N, \tau, (i, f_i)_i\big)$ *et retourne*

$$\big(E_{\mathcal{R}}, E_{\mathcal{AS}}, (E_{\mathcal{C}})_{\mathcal{C}}\big)$$

3) *Pour tout* $j \in \{1, ..., N\}$ *et tout* $f' \in \{0,1\}^k$ *l'algorithme* $\mathcal{R}_1$ *prend en entrée* $(j, f', I_{\mathcal{R}}, E_{\mathcal{R}})$ *et retourne* $(m_{\mathcal{R}_1}, e_{\mathcal{R}_1})$.

4) *Pour tout* $m_{\mathcal{R}_1} \in \mathcal{R}_1(j, f', I_{\mathcal{R}}, E_{\mathcal{R}})$ *l'algorithme* $\mathcal{AS}$ *prend en entrée* $(m_{\mathcal{R}_1}, I_{\mathcal{AS}}, E_{\mathcal{AS}})$ *et interagit avec l'ensemble des composantes* $\mathcal{C}$*, où chaque composante prend en entrée initiale* $(I_{\mathcal{C}}, E_{\mathcal{C}})$ *:*

$$\Big\langle \mathcal{AS}(m_{\mathcal{R}_1}, I_{\mathcal{AS}}, E_{\mathcal{AS}}); \big(\mathcal{C}(I_{\mathcal{C}}, E_{\mathcal{C}})\big)_{\mathcal{C}} \Big\rangle$$

Cette interaction retourne $\big(m_{\mathcal{AS}}, e_{\mathcal{AS}}, (m_{\mathcal{C}}, e_{\mathcal{C}})_{\mathcal{C}}\big)$.

5) *Pour tout* $m_{\mathcal{AS}}$ *provenant de l'interaction du point* ***4)*** *l'algorithme* $\mathcal{R}_2$ *prend en entrée* $(m_{\mathcal{AS}}, e_{\mathcal{R}_1})$ *et calcule puis retourne* $m_{\mathcal{R}_2} \in \{0, 1, \perp\}$.

6) *Enfin, si* (j, f') *vérifie* $hw(f' \oplus f_j) \leq \tau$ *(resp.* $hw(f' \oplus f_j) > \tau$*) et que* $m_{\mathcal{AS}}$ *est calculé comme ci-dessus, on a* $m_{\mathcal{R}_2} = 1$ *(resp.,* $m_{\mathcal{R}_2} = 0$*).*

Il sera commode dans la suite d'adopter la notation suivante : pour $\kappa \in \{0,1\}^k$ et $\tau \in \{0, ..., k\}$, on pose $j(\kappa, \tau) = 1$ si $hw(\kappa) \leq \tau$ et $j(\kappa, \tau) = 0$ si $hw(\kappa) > \tau$. Nous rappellerons cette notation lorsque cela sera nécessaire.

Il convient maintenant d'expliquer le contenu de cette définition. On le fait ci-dessous point par point pour essayer d'être le plus clair possible.

• La fonctionnalité INFRA sert à créer des paramètres globaux pour le système. Ces paramètres doivent être indépendants de la base de données qui sera amenée à s'enregistrer mais on leur donne un paramètre de génération en commun pour spécifier une borne sur la taille de la base de données que peut gérer le système. Les paramètres générés sont alors distribués aux différentes composantes : certains - $I_{\mathcal{R}}$, $I_{\mathcal{AS}}$ et les $I_{\mathcal{C}}$ - sont propres à ces composantes, et d'autres - I_{ext} - sont externes.

• La fonctionnalité ENR enregistre la base de données qu'elle reçoit en entrée. Cela se fait en calculant et en distribuant aux composantes des informations qui leur sont propres : $E_{\mathcal{R}}$, $E_{\mathcal{AS}}$ et les $E_{\mathcal{C}}$.

• Les algorithmes $\mathcal{R}_1$ et $\mathcal{R}_2$ sont moralement le lecteur traitant deux informations différentes. La partie $\mathcal{R}_1$ génère le message initialisant la connection entre le lecteur et le serveur et la partie $\mathcal{R}_2$ détermine le statut final de la ronde d'authentification. Ce dernier peut soit être valide - c'est-à-dire que le protocole s'est déroulé sans problème - auquel cas une réponse (0 ou 1) quand à l'authenticité de l'objet est acceptée, soit invalide auquel cas la ronde est annulée ($\perp$).

• Les interactions entre $\mathcal{AS}$ et les $\mathcal{C}$ sont regroupées pour simplifier. Ces interactions retournent simplement un compte rendu des messages et états calculés par les différentes composantes techniques.

• Le dernier point exprime juste l'exigence selon laquelle on obtient bien une réponse juste si les algorithmes fonctionnent correctement. Autrement dit, si

la ronde est validée, l'objet est considéré comme authentique (respectivement, contrefait) si $m_{\mathcal{R}_2} = 1$ (respectivement, $m_{\mathcal{R}_2} = 0$). Si la ronde n'est pas validée, $m_{\mathcal{R}_2} = \perp$.

Remarques : • L'intérêt de couper $\mathcal{R}$ en deux ne se réduit pas à distinguer deux phases de fonctionnement. Cette distinction nous permet aussi de donner une bonne définition de la garantie d'intégrité des messages lors d'une ronde d'authentification. Il faut séparer la réponse finale de la génération des messages auxquels l'adversaire à droit pour essayer de fausser une ronde.

• Dans la définition précédente, le terme "interagit" est un peu flou. Pour lui donner un sens formel précis il faut utiliser le langage des machines de Turing interactives, ce qui alourdit considérablement la description (pour plus de détails sur les machines de Turing interactives et leur lien avec le la cryptographie, voir [15]). Nous nous contenterons d'indiquer que des machines de Turing interagissent si, comme on s'y attend, elles se passent l'une à l'autre des messages avant de s'arrêter. Bien entendu, toutes les machines ici étant des PTA, les messages transmis d'une machine à une autre sont de longueur polynomiale et en nombre polynomial.

Nous pouvons maintenant donner une définition formelle pour le modèle de sécurité. Il s'agit de construire des **jeux d'attaque** où des PPTA modélisant des adversaires reçoivent en entrée certaines informations provenant :

1) de l'infrastructure du système et

2) des rondes d'authentifications.

Selon le positionnement de l'adversaire par rapport aux composantes du système, ces informations varient. Si l'adversaire est une composante serveur malicieuse, les informations qu'il reçoit sont les messages que reçoit cette composante lors d'une ronde, ainsi que son infrastructure personnelle par exemple.

Nous avons vu plus haut qu'informellement le modèle de sécurité prévoit les différents types d'adversaires suivants :

• des adversaires extérieurs observant les communications entre toutes les composantes du système. Ils cherchent à déterminer l'identité de l'objet qui s'authentifie à l'aide des informations qui circulent entre toutes les composantes lors d'une ronde d'authentification, mais sans avoir accès aux informations internes aux composantes ou calculées par elles et non communiquées ;

• des adversaires extérieurs modifiant les communications entre le lecteur $\mathcal{R}$ et le serveur d'authentification $\mathcal{AS}$. Ils cherchent à modifier la réponse finale de l'authentification sans que le système ne s'en rende compte ;

• les composantes $\mathcal{AS}$ et $\mathcal{C}$ elles-mêmes, qui cherchent à déterminer l'identité de l'objet qui s'authentifie. Ils ont à leur disposition leurs données dans l'infrastructure, mais pas celles des autres composantes, et les informations qu'ils reçoivent lors d'une tentative d'authentification.

Formellement, les jeux qu'il faut définir pour codifier le comportement de

ces adversaires sont les suivants, où $n \in \mathbb{N}$ est un paramètre de sécurité :

$$
\begin{aligned}
&\mathbf{Exp}^{oa-ext}_{\mathcal{PAM},\mathcal{A}}(n)\\
&\big(I_{ext}, I_{\mathcal{R}}, I_{\mathcal{AS}}, (I_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \textsc{Infra}(1^n)\\
&\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, e\Big) \leftarrow \mathcal{A}_1(I_{ext})\\
&\big(E_{\mathcal{R}}, E_{\mathcal{AS}}, (E_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \textsc{Enr}\big(k, N, \tau, (i, f_i)_i\big)\\
&\beta \leftarrow \{0,1\}\\
&(m_{\mathcal{R}_1}, e_{\mathcal{R}_1}) \leftarrow \mathcal{R}_1(i_\beta, f_{i_\beta} \oplus \kappa, I_{\mathcal{R}}, E_{\mathcal{R}})\\
&\big(m_{\mathcal{AS}}, e_{\mathcal{AS}}, (m_{\mathcal{C}}, e_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \Big\langle \mathcal{AS}(m_{\mathcal{R}_1}, I_{\mathcal{AS}}, E_{\mathcal{AS}}), \big(\mathcal{C}(I_{\mathcal{C}}, E_{\mathcal{C}})\big)_{\mathcal{C}} \Big\rangle\\
&m_{\mathcal{R}_2} \leftarrow \mathcal{R}_2(m_{\mathcal{AS}}, e_{\mathcal{R}_1})\\
&\gamma \leftarrow \mathcal{A}_2\big(m_{\mathcal{R}_1}, m_{\mathcal{AS}}, (m_{\mathcal{C}})_{\mathcal{C}}, e\big)\\
&\text{Return } \gamma
\end{aligned}
$$

où on définit l'avantage de l'adversaire $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ comme étant

$$Adv^{oa-ext}_{\mathcal{PAM},\mathcal{A}}(n) := \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|,$$

$$
\begin{aligned}
&\mathbf{Exp}^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}(n)\\
&\big(I_{ext}, I_{\mathcal{R}}, I_{\mathcal{AS}}, (I_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \textsc{Infra}(1^n)\\
&\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, e\Big) \leftarrow \mathcal{CM}_1(I_{ext}, I_{\mathcal{CM}})\\
&\big(E_{\mathcal{R}}, E_{\mathcal{AS}}, (E_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \textsc{Enr}\big(k, N, \tau, (i, f_i)_i\big)\\
&\beta \leftarrow \{0,1\}\\
&(m_{\mathcal{R}_1}, e_{\mathcal{R}_1}) \leftarrow \mathcal{R}_1(i_\beta, f_{i_\beta} \oplus \kappa, I_{\mathcal{R}}, E_{\mathcal{R}})\\
&\big(m_{\mathcal{AS}}, e_{\mathcal{AS}}, (m_{\mathcal{C}}, e_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \Big\langle \mathcal{AS}(m_{\mathcal{R}_1}, I_{\mathcal{AS}}, E_{\mathcal{AS}}), \big(\mathcal{C}(I_{\mathcal{C}}, E_{\mathcal{C}})\big)_{\mathcal{C}} \Big\rangle\\
&m_{\mathcal{R}_2} \leftarrow \mathcal{R}_2(m_{\mathcal{AS}}, e_{\mathcal{R}_1})\\
&\gamma \leftarrow \mathcal{CM}_2\big(m_{\mathcal{CM}}, e_{\mathcal{CM}}, e\big)\\
&\text{Return } \gamma
\end{aligned}
$$

où on définit l'avantage de l'adversaire $\mathcal{CM} = (\mathcal{CM}_1, \mathcal{M}_2) \in \{\mathcal{AS}\} \cup \{\mathcal{C}\}_{\mathcal{C}}$ comme étant

$$Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}(n) := \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|$$

et

$$
\begin{aligned}
&\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{A}}(n)\\
&\big(I_{ext}, I_{\mathcal{R}}, I_{\mathcal{AS}}, (I_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \textsc{Infra}(1^n)\\
&\big((m_{\mathcal{R}_1 o}, m_{\mathcal{AS} o}, m_{\mathcal{R}_2 o})_o, e_1\big) \leftarrow \mathcal{A}_1\Big(\mathcal{OR}_{I_{ext}, I_{\mathcal{R}}, I_{\mathcal{AS}}, (I_{\mathcal{C}})_{\mathcal{C}}}\Big)\\
&\Big(j, \big(k, N, \tau, (i, f_i)_i\big), \kappa, e_2\Big) \leftarrow \mathcal{A}_2\big(I_{ext}, (m_{\mathcal{R}_1 o}, m_{\mathcal{AS} o}, m_{\mathcal{R}_2 o})_o, e_1\big)\\
&\big(E_{\mathcal{R}}, E_{\mathcal{AS}}, (E_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \textsc{Enr}\big(k, N, \tau, (i, f_i)_i\big)\\
&(m_{\mathcal{R}_1}, e_{\mathcal{R}_1}) \leftarrow \mathcal{R}_1(j, f_j \oplus \kappa, I_{\mathcal{R}}, E_{\mathcal{R}})\\
&(cm_{\mathcal{R}_1}, e_3) \leftarrow \mathcal{A}_3(m_{\mathcal{R}_1}, e_2)\\
&\big(cm^1_{\mathcal{AS}}, ce_{\mathcal{AS}}, (cm_{\mathcal{C}}, ce_{\mathcal{C}})_{\mathcal{C}}\big) \leftarrow \Big\langle \mathcal{AS}(cm_{\mathcal{R}_1}, I_{\mathcal{AS}}, E_{\mathcal{AS}}), \big(\mathcal{C}(I_{\mathcal{C}}, E_{\mathcal{C}})\big)_{\mathcal{C}} \Big\rangle\\
&(cm^2_{\mathcal{AS}}) \leftarrow \mathcal{A}_4(cm^1_{\mathcal{AS}}, e_3)\\
&cm_{\mathcal{R}_2} \leftarrow \mathcal{R}_2(cm^2_{\mathcal{AS}}, e_{\mathcal{R}_1})\\
&\text{Return } cm_{\mathcal{R}_2}
\end{aligned}
$$

où on définit l'avantage de $\mathcal{A}$ comme étant :

$$Adv^{int-ext}_{\mathcal{PAM},\mathcal{A}}(n) := \mathbb{P}[cm_{\mathcal{R}_2} = j(\kappa, \tau) \oplus 1]$$

Définition 3 • *On dit que le schéma d'authentification $\mathcal{PAM}$* ***garantit l'anonymat d'objet contre les adversaires extérieurs*** *si pour tout PPTA $\mathcal{A}$ l'application*

$$Adv^{oa-ext}_{\mathcal{R},\mathcal{AS},(\mathcal{C})_{\mathcal{C}},\mathcal{A}}$$

est négligeable en le paramètre de sécurité.

• *On dit que le schéma d'authentification $\mathcal{PAM}$* ***garantit l'anonymat d'objet contre les composantes serveur malicieuses*** *si pour tout PPTA $\mathcal{CM} \in \{\mathcal{AS}\} \cup \{\mathcal{C}\}_{\mathcal{C}}$ l'application*

$$Adv^{oa-comp}_{\mathcal{R},\mathcal{AS},(\mathcal{C})_{\mathcal{C}},\mathcal{A}}$$

est négligeable en le paramètre de sécurité.

• *On dit que le schéma d'authentification $\mathcal{PAM}$* ***garantit l'intégrité de la réponse d'une ronde d'authentification contre les adversaires extérieurs*** *si pour tout PPTA $\mathcal{A}$ l'application*

$$Adv^{int-ext}_{\mathcal{PAM},\mathcal{A}}$$

est négligeable en le paramètre de sécurité.

Rappelons qu'on dit d'une application $g : \mathbb{N} \to \mathbb{R}_+$ qu'elle est *négligeable* si pour tout polynôme positif p (i.e. pour tout $r \in \mathbb{R}$, $p(r) > 0$), si $n \in \mathbb{N}$ est suffisamment grand on a

$$g(n) \leq \frac{1}{p(n)}$$

Avant de poursuivre, expliquons le contenu de ces jeux d'attaque ligne par ligne :

$\mathbf{Exp}^{oa-ext}_{\mathcal{PAM},\mathcal{A}}$

La première phase est une phase de mise en place des paramètres globaux du système faisant appel à INFRA avec le paramètre de sécurité en entrée. Cette phase est d'ailleurs commune aux trois jeux. On demande ensuite à la première partie de l'attaquant, $\mathcal{A}_1$, de générer une base de données et de choisir deux indices i_0 et i_1 correspondant à des flux de bits distincts dans la base choisie. La base est ensuite enregistrée au sein du système et les composantes reçoivent de l'information relative à cette base de données particulière. La phase suivante est la phase du challenge, où l'un des deux indices est choisi uniformément au hasard et une ronde d'authentification est effectuée. L'adversaire étant extérieur, il reçoit comme information les messages $\left(m_{\mathcal{R}_1}, m_{\mathcal{AS}}, (m_{\mathcal{C}})_{\mathcal{C}}\right)$ envoyés par toutes les composantes entre elles. Enfin, avec ces messages $\mathcal{A}_2$ doit deviner quel indice a été choisi.

$\mathbf{Exp}^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$

Ce jeu est presque identique au premier. La seule différence est l'ensemble des messages passé à l'adversaire après que la ronde d'authentification ait été faite : comme c'est une composante malicieuse, les messages auxquels elle a droit sont ceux qui lui parviennent. Elle dispose en plus de l'information relative à sa propre infrastructure et inconnue d'un adversaire extérieur et des autres composantes.

$\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{A}}$

Enfin, le troisième jeu est celui modélisant l'attaque d'un adversaire extérieur voulant modifier la réponse finale. La première phase est celle de la mise en place des paramètres comme pour les deux autres jeux. La deuxième phase est une phase de **requêtes** de rondes d'authentification. Précisément, $\mathcal{A}_2$ dispose d'un oracle

$$\mathcal{OR}_{I_{ext},I_{\mathcal{R}},I_{\mathcal{AS}},(I_{\mathcal{C}})_{\mathcal{C}}}$$

auquel il peut faire appel pour obtenir des messages de rondes d'authentification valides avec en entrée des bases de données de son choix. Cela est dû au fait que $\mathcal{A}$ soit un adversaire actif. Dans l'étape suivante, $\mathcal{A}_3$ choisit, en connaissant les réponses de l'oracle, une base de données et un indice de cette base. Enfin, il a pour tâche de modifier avec succès la réponse lors d'une ronde d'authentification portant sur l'indice et la base choisie.

Remarque importante sur l'intégrité de la réponse finale : Le modèle de sécurité choisi concernant l'intégrité de la réponse finale ne capture pas des adversaires qui chercheraient simplement à faire échouer des rondes d'authentification. Dans les protocoles qui sont présentés au chapitre 4, il est en particulier possible, en rejouant des messages déjà envoyés, de désynchroniser les compteurs, obligeant toute tentative d'authentification postérieure à cette manipulation à retourner le symbole d'échec.

Remarques :

• Le générateur de bases de données n'a aucun rôle dans les définitions ci-dessus. Cela est dû au fait que nous nous plaçons, conformément à une pratique standard, dans un cadre de sécurité théorique où l'attaquant à un contrôle sur les messages à protéger. Les détails concernant la sécurité théorique se trouvent dans le chapitre 4, mais l'idée ici est que la confidentialité et l'intégrité d'une authentification sont garanties même si l'objet à authentifier a un flux morphométrique choisi par l'adversaire.

• On ne demande pas l'intégrité de la totalité des communications d'une ronde. La définition contient juste assez de contraintes pour que lors d'une ronde d'authentification un adversaire ne puisse pas changer la réponse finale sans être détecté. En pratique, c'est ce qui nous intéresse : peu importe que l'adversaire ait pu changer les communications. Si la réponse finale est la bonne, l'authentification s'est faite quand même.

Chapitre 2

Eléments de cryptographie

Dans ce chapitre nous faisons les rappels mathématiques et cryptographiques nécessaires pour décrire les protocoles et analyser leur sécurité. Celle-ci sera prouvée dans le modèle formel de sécurité défini à la fin du chapitre précédent.

L'un de nos protocoles utilise un schéma de chiffrement obtenu en modifiant le schéma ElGamal classique ; cela est fait car nous avions en tête l'utilisation de courbes elliptiques. Nous commençons donc par rappeler un peu comment de telles courbes trouvent leur place en cryptographie. Ensuite, nous rappelons les définitions des primitives cryptographiques dont nous aurons besoin pour construire nos protocoles, en insistant sur les schémas de chiffrement pour lesquels une nouvelle notion de sécurité est définie et caractérisée au chapitre 3. Nous rappelons le schéma de chiffrement ElGamal, définissons celui que nous utiliserons pour l'un de nos protocoles et nous prouverons qu'il est *ind-cpa*-sécurisé. Les deux autres primitives, rappelées bien plus brièvement, sont les schémas de signature digitale et les protocoles de récupération de données.

2.1 Courbes elliptiques sur les corps finis

Nous faisons ici un rapide rappel sur les courbes elliptiques définis sur un corps fini. Pour plus de détails concernant les courbes elliptiques en général, voir [35]. Pour les applications en cryptographie, voir [20], [2] ou [3].

2.1.1 Courbes elliptiques

Dans tout ce paragraphe, on se fixe un corps commutatif K et une clôture algébrique $\overline{K}$ de K.

Définition

Une *équation de Weierstrass sur* K est une équation E de la forme

$$E : \ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

où les a_i sont tous dans K. Le discriminant Δ de E est défini comme suit :

$$\begin{aligned} d_2 &:= a_1^2 + 4a_2 \\ d_4 &:= 2a_4 + a_1a_3 \\ d_6 &:= a_3^2 + 4a_6 \\ d_8 &:= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \\ \Delta &:= -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \end{aligned}$$

Définition 4 *On dit qu'une équation de Weierstrass E comme ci-dessus est l'équation d'une* ***courbe elliptique*** *si $\Delta \neq 0$.*

Dorénavant, nous supposerons toujours que $\Delta \neq 0$, i.e. lorsqu'une équation de Weierstrass sera donnée, ce sera toujours l'équation d'une courbe elliptique. Nous parlerons d'une courbe elliptique définie avec les coefficients a_i plutôt qu'une courbe elliptique dont une équation de Weierstrass est donnée par les coefficients a_i.

Soit E une courbe elliptique définie avec les coefficients a_i. On dit que E est définie sur K, et on note dans ce cas E/K, quand les a_i sont dans K.

Définition 5 *Soit E/K une courbe elliptique donnée par les coefficients a_i (donc $a_i \in K$) et soit L une extension de K. L'ensemble des* ***points L-rationnels*** *de E est*

$$E(L) := \left\{(x,y) \in L^2 \;\middle|\; y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\right\} \cup \{\infty\}$$

Autrement dit, les points L-rationnels de E/K sont les solutions dans L de l'équation qui définit E, et un point spécial noté ∞. Le point ∞ est appelé *le point à l'infini de E/K*. Il est considéré comme L-rationnel pour toute extension L de K.

Isomorphismes de courbes elliptiques, forme courte de Weierstrass

Soient E/K et E'/K deux courbes elliptiques (définies sur K) données respectivement par les coefficients a_i et a'_i, i.e.

$$E: \;\; y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

et

$$E': \;\; y^2 + a'_1xy + a'_3y = x^3 + a'_2x^2 + a'_4x + a'_6$$

Définition 6 *Soit L une extension de K. On dit que E et E' sont* ***isomorphes sur L,*** *ou L-isomorphes, s'il existe $(u,r,s,t) \in L^4$, $u \neq 0$, tel que le changement de coordonnées*

$$(x,y) \to (u^2x + r, u^3y + u^2sx + t)$$

transforme l'équation de E en celle de E'. Un tel changement de coordonnées est dit ***admissible.***

Deux courbes isomorphes sur K ont la même structure sur K. On peut donc restreindre l'étude d'une classe de K-isomorphisme de courbes à celle de l'une de ces courbes. En choisissant un bon changement de coordonnées admissible, on simplifie considérablement l'équation avec laquelle on travaille. Cependant, les changements de coordonnées permettant cette simplification changent en fonction de la caractéristique de K. Spécifiquement, il faut considérer séparément la caractéristique $p > 3$ des caractéristiques $p = 2$ et $p = 3$. Nous nous bornerons à travailler en caractéristiques $p > 3$ et $p = 2$, le cas $p = 3$ étant bien moins utilisé en cryptographie.

- **Si** $p > 3$

  Le changement de coordonnées admissible

$$(x, y) \to \left(\frac{x - 3a_1^2 - 12a_2}{36}, \frac{y - 3a_1 x}{216} - \frac{a_1^3 + 4a_1 a_2 - 12a_3}{24} \right)$$

transforme la courbe E en une courbe de la forme

$$y^2 = x^3 + ax + b$$

avec a et b dans K. Le discriminant de cette courbe est $\Delta = -16(4a^3 + 27b^2)$.

- **Si** $p = 2$

  Si $a_1 \neq 0$, le changement de coordonnées admissible

$$(x, y) \to \left(a_1^2 x + \frac{a_3}{a_1}, a_1^3 y + \frac{a_1^2 a_4 + a_3^2}{a_1^3} \right)$$

transforme E en une courbe donnée par

$$y^2 + xy = x^3 + ax^2 + b$$

avec a et b dans K. Son discriminant est $\Delta = b$.

  Si $a_1 = 0$, le changement de coordonnées admissible

$$(x, y) \to (x + a_2, y)$$

transforme E en une courbe de la forme

$$y^2 + cy = x^3 + ax + b$$

avec a, b et c dans K et discriminant $\Delta = c^4$.

Dorénavant, toutes les courbes que nous considèrerons auront l'une des trois formes ci-dessus.

Loi de groupe sur une courbe elliptique

Soit E/K une courbe elliptique. Il est possible de définir une loi de groupe abélien sur E. Elle se comprend le mieux de manière géométrique, en prenant pour corps de base $\mathbb{R}$ (voir la figure 1 ci-dessous). On peut donc supposer que l'équation de E est de la forme $y^2 = x^3 + ax + b$ $(*)$ avec a et b dans $\mathbb{R}$ (et $\Delta \neq 0$).

Soient P et Q deux points de E que l'on va supposer distincts pour l'instant. La théorie générale montre que l'unique droite D passant par P et Q coupe la courbe E en un troisième point R' ou bien est parallèle à l'axe des ordonnées (l'axe donné par $x = 0$). Plaçons nous dans le premier cas pour l'instant et soit R le point du plan symétrique de R' par rapport à l'axe des abscisses ($y = 0$). Il se trouve que $R \in E(\mathbb{R})$, i.e. que les coordonnées (x_R, y_R) de R satisfont l'équation $(*)$. Ce point est défini comme étant la somme de P et Q. On note

$$R = P + Q$$

Qu'en est-il du deuxième cas ? Si D est parallèle à l'axe des ordonnées, on dit simplement que D intersecte E à l'infini, c'est-à-dire en ∞. Ce point est en fait l'élément neutre de la loi de groupe. On peut ainsi également décrire une procédure pour calculer l'inverse d'un point $P \in E$ distinct de ∞. L'unique droite D passant par P et ∞ est l'unique droite D passant par P et parallèle à l'axe des ordonnées. A nouveau, la théorie générale montre que D coupe E un unique troisème point qui se trouve être l'inverse de P. Comme il est d'usage de noter la loi additivement, on parle d'opposé plutôt que d'inverse.

Enfin, pour doubler un point P, la procédure est la même que pour additionner deux points distincts, mais en prenant pour D la tangente à E en P.

Ces procédures peuvent s'exprimer algébriquement en fonction des équations de la courbe et des droites que l'on considère. D'une part cela fournit des expressions concrètes pour les coordonnées de la somme de deux points en fonction de celles des deux points en question, ainsi que les coordonnées des inverses, et d'autre part cela montre que la loi de groupe existe sur n'importe quel corps de base. En particulier, on obtient un joli groupe fini si K est un corps fini.

Formules explicites

Soient E/K une courbe elliptique où le corps de base est à nouveau quelconque, et P et Q dans $E(K)$. On note

$$P = (x_P, y_P) \text{ et } Q = (x_Q, y_Q)$$

Si K est de caractéristique $p > 3$

La courbe est de la forme $y^2 = x^3 + ax + b$ avec a et b dans K.

- **Elément neutre**

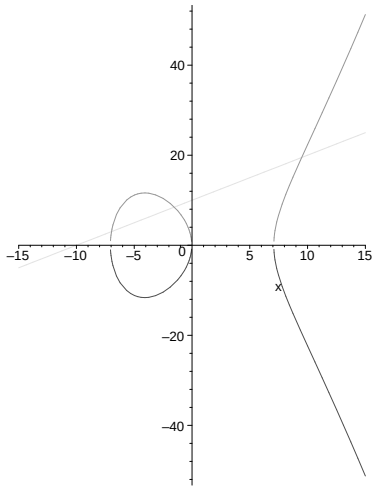


FIGURE 2.1 – Courbe elliptique d'équation $y^2 = x^3 - 50x - 1$ vue dans $\mathbb{R}^2$

$P \oplus \infty = \infty \oplus P = P$, c'est-à-dire que le point à l'infini est le neutre pour la loi d'addition ;

- **Opposé d'un élément**

  $-P = (x_P, -y_P)$, autrement dit l'opposé de P est son symétrique par rapport à l'axe des abscisses ;

- **Addition de deux points distincts, non neutres, non opposés**

  on a $P \oplus Q = (x_{P+Q}, y_{P+Q})$ où

$$x_{P+Q} := \left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q$$

et

$$y_{P+Q} := \frac{y_Q - y_P}{x_Q - x_P}(x_P - x_{P+Q}) - y_P$$

- **Doublement d'un point non neutre**

  on a $2P = (x_{2P}, y_{2P})$ où

$$x_{2P} := \frac{3x_P^2 + a}{2y_P} - 2x_P$$

et

$$y_{2P} = \frac{3x_P^2 + a}{2y_P}(x_P - x_{2P}) - y_P$$

Si K est de caractéristique $p = 2$

Il y a deux cas, libellés i) et ii) ci-dessous, à considérer selon l'équation réduite utilisée.

i) Si la courbe est de la forme $y^2 + xy = x^3 + ax^2 + b$, a et b étant dans K :

- **Elément neutre**
  $P \oplus \infty = \infty \oplus P = P$, c'est-à-dire que le point à l'infini est le neutre pour la loi d'addition ;

- **Opposé d'un élément**
  on a $-P = (x_P, x_P + y_P)$;

- **Addition de deux points distincts, non neutres, non opposés**
  on a $P \oplus Q = (x_{P+Q}, y_{P+Q})$ où

$$x_{P+Q} := \left(\frac{y_P + y_Q}{x_P + x_Q}\right)^2 + \frac{y_P + y_Q}{x_P + x_Q} + x_P + x_Q + a$$

et

$$y_{P+Q} := \frac{y_P + y_Q}{x_P + x_Q}(x_P + x_{P+Q}) + x_{P+Q} + y_P$$

- **Doublement d'un point non neutre**
  on a $2P = (x_{2P}, y_{2P})$ où

$$x_{2P} := \left(\frac{x_P + y_P}{x_P}\right)^2 + \frac{x_P + y_P}{x_P} + a$$

et

$$y_{2P} := x_P^2 + \frac{x_P + y_P}{x_P} x_{2P} + x_{2P}$$

ii) Si la courbe est de la forme $y^2 + cy = x^3 + ax + b$, a, b et c étant dans K :

- **Elément neutre**
  $P \oplus \infty = \infty \oplus P = P$, c'est-à-dire que le point à l'infini est le neutre pour la loi d'addition ;

- **Opposé d'un élément**
  on a $-P = (x_P, y_P + c)$;

- **Addition de deux points distincts, non neutres, non opposés**
  on a $P \oplus Q = (x_{P+Q}, y_{P+Q})$ où

$$x_{P+Q} := \left(\frac{y_P + y_Q}{x_P + x_Q}\right)^2 + x_P + x_Q$$

et

$$y_{P+Q} := \frac{y_P + y_Q}{x_P + x_Q}(x_P + x_{P+Q}) + y_P + c$$

- **Doublement d'un point non neutre**

  on a $2P = (x_{2P}, y_{2P})$ où

$$x_{2P} := \left(\frac{x_P^2 + a}{c}\right)^2$$

et

$$y_{2P} = \frac{x_P^2 + a}{c}(x_P + x_{2P}) + y_P + c$$

On conclut par le théorème suivant :

Théorème 1 *Soit E/K une courbe elliptique ayant l'une des formes ci-dessus, et L une extension de K. L'opération d'addition définie par les formules précédentes fait de $E(L)$ un groupe abélien. En particulier, $E(K)$ et $E(\overline{K})$ sont des groupes abéliens.*

2.1.2 Courbes elliptiques sur les corps finis

Dans ce paragraphe nous prenons $K = \mathbb{F}_p$ ou $K = \mathbb{F}_{2^m}$ où $p > 3$ et $m \in \mathbb{N}$, $m \geq 2$, i.e. nous nous plaçons soit sur un corps fini premier de caractéristique distincte de 3, soit sur un corps fini de caractéristique 2. Nous noterons q le cardinal de K, donc $q = p$ si $K = \mathbb{F}_p$ et $q = 2^m$ si $K = \mathbb{F}_{2^m}$.

Soit E/K une courbe elliptique. Il est clair que K étant fini, $E(K)$ est fini. Il résulte de ceci et des résultats résumés ci-dessus que l'ensemble $E(K)$ des points K-rationnels de E est un groupe *fini.* Le théorème de Hasse ci-dessous donne une borne sur le nombre de points K-rationnels de E.

Théorème 2 *Le nombre $\big|E(K)\big|$ de points K-rationnels de E vérifie*

$$\Big|\big|E(K)\big| - (q+1)\Big| \leq 2\sqrt{q}$$

Quand K devient grand, il y a donc pas mal de points ! En faisant varier la courbe et le corps, on arrive à trouver des courbes pour lesquelles $E(K)$ est d'ordre un grand nombre premier du même ordre de grandeur que q.

2.1.3 Utilisation en cryptographie

Historiquement, les courbes elliptiques sont apparues pour la première fois en cryptographie dans les années 80 où dans [25] Lenstra propose un algorithme de factorisation d'entiers basé sur ces courbes. C'est une adaptation au cas d'une courbe elliptique de l'algorithme $p - 1$ de Pollard (voir [32]), l'avantage étant

que le fait de pouvoir faire varier la courbe permet de faire varier le groupe sous-jacent à l'algorithme.

Ensuite, c'est en 1985 que Koblitz et Miller (dans [22] et [29] respectivement) proposent d'utiliser le groupe des points rationnels d'une courbe elliptique définie sur un corps fini pour cette fois développer des primitives cryptographiques basées sur le problème du logarithme discret (DLP) dans ce groupe. L'idée s'est révélée fructueuse en particulier parce qu'à ce jour, aucun algorithme sous-exponentiel n'a encore été trouvé pour résoudre le DLP elliptique. Ce n'est pas le cas du problème de la factorisation d'un entier (dont dépend RSA) ou du DLP classique dans le groupe multiplicatif d'un corps fini (dont dépend par exemple le *digital signature algorithm*), pour lesquels le crible sur corps de nombres ([31], [26] et [18]) a été développé. Il en résulte qu'à sécurité égale, les méthodes elliptiques demandent des paramètres moins grands, et la différence est d'autant plus prononcée lorsque la sécurité croît. Le tableau ci-dessous compare la sécurité d'un système basé sur le DLP d'une courbe elliptique avec celle d'un système basé sur la factorisation d'un entier RSA. La première ligne contient le niveau de sécurité atteint (avec nos connaissances actuelles en termes d'algorithmes pour factoriser des entiers et trouver des logarithmes discrets dans des courbes elliptiques), au sens où il faut en moyenne 2^x opérations pour casser le système au niveau x. Les deuxième et troisième lignes donnent les tailles de l'ordre du groupe utilisé et de l'entier modulo lequel on travaille pour des systèmes utilisant une courbe elliptique et un entier RSA respectivement.

sécurité	80	112	128	192	256
EC	160	224	256	384	512
RSA	1248	2432	3248	7936	15424

Il faut quand même prendre garde aux paramètres choisis pour la courbe. En effet, certaines courbes présentent des faiblesses : certaines d'entre elles ont un DLP facile à résoudre (par exemple [36]), d'autres ont un DLP supposé difficile à résoudre mais où le problème décisionnel de Diffie-Hellman (DDH, voir le paragraphe 2.2.2 ci-dessous) est facile à résoudre (des groupes qui séparent les deux problèmes de cette façon sont exhibés dans [21] par exemple). Or, on le verra un peu plus loin, la difficulté de DDH est cruciale pour sécuriser en un sens très fort certaines primitives cryptographiques.

A titre indicatif, on reproduit ici les hypothèses de base qui sont faites sur les courbes dans [20]. Pour K égal à $\mathbb{F}_q$ (avec les notations des paragraphes précédents), soit E/K une courbe elliptique et $\mathbb{G} := E(K)$ le groupe de ses points K-rationnels.

• L'ordre $|\mathbb{G}|$ de $\mathbb{G}$ doit être de la forme ew où $e \in \{1, 2, 3, 4\}$ et w est un nombre premier de longueur binaire au moins égale à 160. Cela rend impraticable les meilleurs algorithmes génériques pour calculer des logarithmes discrets (méthodes ρ de Pollard, méthode de Pohlig-Hellman).

• Il faut prendre garde à ne pas choisir une courbe telle que $|\mathbb{G}| = q$ (attaque de Smart [36]).

• Il faut que w ne divise pas $q^k - 1$ pour des entiers naturels k trop petits. En pratique, il faut que w ne divise pas $q^k - 1$ pour k valant jusqu'à 20. Cette précaution rend les attaques via les pairings de Weil et Tate (voir [28] et [13] par exemple) impraticables. Notamment, DDH reste à priori difficile.

• Enfin, pour éviter des attaques exploitant la technique de descente de Weil (voir [14] et [19]), lorsque q est de la forme 2^m, il faut que m soit premier.

Mentionnons qu'il est possible de générer des courbes ayant ces propriétés de manière vérifiablement aléatoire.

Pour une bonne synthèse de l'histoire de la cryptographie sur courbe elliptique, le lecteur peut consulter [23]. S'y trouve également une discussion portant sur d'autres choix à faire sur les paramètres.

2.1.4 Un exemple concret : ECIES

L'ECIES (Elliptic Curve Intergrated Encryption Scheme) est un schéma de chiffrement standard utilisant les courbes elliptiques. C'est un exemple de ce qui est appelé un schéma hybride, c'est-à-dire mélangeant des techniques de chiffrement symétrique et asymétrique. Nous le décrivons brièvement ici car nous souhaitons l'utiliser pour certains de nos protocoles d'authentification d'objets.

Conservons les notations des paragraphes précédents : soient K un corps fini de cardinal noté q (égal à p ou a 2^m) et E/K une courbe elliptique convenable pour des applications cryptographiques. Notons $\mathbb{G}$ le groupe $E(K)$ des points $K-$rationnels de E et supposons que $\mathbb{G}$ soit d'ordre un grand nombre premier $q_{\mathbb{G}}$. Soit G un générateur de $\mathbb{G}$.

Donnons nous les primitives cryptographiques suivantes :

• les algorithmes de chiffrement $\mathcal{E}$ et de déchiffrement $\mathcal{D}$ d'un schéma de chiffrement chiffrement symmétrique. Si k est une clef pour ces algorithmes, nous noterons $\mathcal{E}_k(m)$ un chiffrement du message m et $\mathcal{D}_k(c)$ le déchiffrement du chiffré c ;

• un code d'authentification de message (Message Authentication Code)

$$MAC : \{0,1\}^n \times \{0,1\}^* \longrightarrow \{0,1\}^m$$

où n et m sont deux entiers naturels. Cette primitive fonctionne comme une fonction de hachage prenant en entrée une clef secrète en plus du message ;

• une fonction de dérivation de clefs

$$KD : \mathbb{G} \times \mathbb{N} \longrightarrow \{0,1\}^*$$

où le deuxième argument de la fonction donne la longueur de la clef à retourner.

Le destinataire du message détient une clef secrète $x \in \{0, ..., q_{\mathbb{G}} - 1\}$ et comme clef publique $Y := xG$.

Pour chiffrer un message m avec la clef de chiffrement Y, ECIES fonctionne comme suit :

1) On choisit $u \in \{1, ..., q_{\mathbb{G}} - 1\}$ uniformément au hasard ;

2) On calcule $U := uG$ et $T := uY$;

3) On invoque $KD(T, l)$ pour obtenir $k_1||k_2$ où k_1 est une clef secrète pour $\mathcal{E}$ et $\mathcal{D}$ et k_2 est une clef secrète pour MAC ;

4) On chiffre m en invoquant $\mathcal{E}_{k_1}(m)$ pour obtenir c ;

5) On calcule le code d'authentification du chiffré c en invoquant $MAC_{k_2}(c)$ pour obtenir r ;

6) On retourne le chiffré ECIES suivant pour m : (U, c, r).

Le message est chiffré à l'aide d'un algorithme de chiffrement symétrique sous une clef dérivée du point T de la courbe. Le code r fait en sorte que ce schéma soit sécurisé contre des attaques adaptatives à chiffrements choisis.

Pour déchiffrer le chiffré c, ECIES fonctionne comme suit :

1) Le point T est calculé en faisant $T = xU$;

2) Les clefs $k_1||k_2$ sont calculées en faisant $k_1||k_2 = KD(T, l)$;

3) On calcule $r' = MAC_{k_2}(c)$; si $r' \neq r$, le chiffré est considéré come invalide et on retourne un message d'erreur ;

4) Si $r' = r$, on déchiffre c en calculant $m = \mathcal{D}_{k_1}(c)$.

2.1.5 Implémentation

Il existe quelques librairies utilisables en C pour pouvoir faire des calculs sur courbes elliptiques. Il y a PARI/gp, développé pour faire de la théorie algorithmique des nombres. Bien que ce ne soit pas spécifiquement tourné vers la cryptographie, il n'est pas difficile d'implémenter certaines des primitives vues ici, comme ElGamal ou ElGamal bit-à-bit avec des courbes utiles dans le monde réel. Le site internet du projet est http ://pari.math.u-bordeaux.fr/

Plus tournée vers la cryptographie, il y a le projet MIRACL développé par Shamus Software, voir http ://www.shamus.ie/index.php ?page=elliptic-curves.

On mentionne aussi le projet Palo-Alto à Grenoble. Il s'agit d'un développement multi-processeur et multi-coeur pour l'adaptation des logiciels, ainsi que d'une librairie spécifique aux courbes elliptiques et hyperelliptiques : MPHELL.

2.2 Chiffrement

La notion de schéma de chiffrement est la plus classique des primitives cryptographiques. Nous suivons les auteurs de [1] pour en donner une définition, puis nous rappelons brièvement la notion d'indistinguabilité de chiffrements sous une attaque à messages clairs choisis. Cette définition sera reprise et complétée au chapitre 3.

2.2.1 Schémas de chiffrement

Définition 7 *Un* ***schéma de chiffrement*** $\mathcal{CS}$ *est un quadruplet de PPTA*

$$(\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$$

vérifiant les propriétés suivantes.

• *Pour tout* $n \in \mathbb{N}$, $\mathcal{P}$ *prend en entrée* 1^n *et retourne* I, *l'information globale de* $\mathcal{CS}$. *Cette information globale spécifie la structure globale du système et contient le paramètre de sécurité.*

• *Pour tout* $I \in \mathcal{P}(1^n)$, $\mathcal{K}$ *prend en entrée* I *et retourne une paire de clefs* (pk, sk).

• *Pour tout* $(pk, sk) \in \mathcal{K}(I)$ *et tout message* m, $\mathcal{E}$ *prend en entrée* (pk, m) *et retourne un chiffré* c. *On note* $\mathcal{E}_{pk}(m)$ *plutôt que* $\mathcal{E}(pk, m)$.

• *Pour tout* $c \in \mathcal{E}_{pk}(m)$, $\mathcal{D}$ *prend en entrée* (sk, c) *et retourne* m. *On note* $\mathcal{D}_{sk}(c)$ *plutôt que* $\mathcal{D}(sk, c)$.

• *Pour tout* $(pk, sk) \in \mathcal{K}(I)$ *et tout message* m *on a*

$$\mathcal{D}_{sk}\big(\mathcal{E}_{pk}(m)\big) = m$$

Rappelons maintenant une des nombreuses définitions de sécurité d'un schéma de chiffrement, celle que nous allons utiliser. Nous y reviendrons beaucoup plus en détail au chapitre suivant mais nous l'évoquons ici pour pouvoir démontrer qu'un chiffrement que nous allons définir un peu plus loin la satisfait.

Soit $\mathcal{CS} = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ un schéma de chiffrement et soit $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ un PPTA opérant en deux temps. Pour tout $n \in \mathbb{N}$, on considère le jeu d'attaque suivant :

$$\begin{aligned}
&\mathbf{Exp}^{IND-CPA}_{\mathcal{CS},\mathcal{A}}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (sk, pk) \leftarrow \mathcal{K}(I)\\
&\qquad (m_0, m_1, e) \leftarrow \mathcal{A}_1(I, pk)\\
&\qquad \beta \leftarrow \{0, 1\}\\
&\qquad c_\beta \leftarrow \mathcal{E}_{pk}(m_\beta)\\
&\qquad \gamma \leftarrow \mathcal{A}_2(m_0, m_1, e, c_\beta)\\
&\quad \text{Return } \gamma
\end{aligned}$$

où on pose

$$Adv^{IND-CPA}_{\mathcal{CS},\mathcal{A}}(n) := \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|$$

Pour éviter des distinctions trivialement possibles il est supposé ci-dessus que les deux messages retournés par $\mathcal{A}_1$ sont de même longueur. On a alors la

Définition 8 *On dit que* $\mathcal{CS}$ *est* IND-CPA-***sécurisé*** *si pour tout PPTA* $\mathcal{A}$ *comme ci-dessus l'application* $Adv^{IND-CPA}_{\mathcal{CS},\mathcal{A}}$ *est négligeable.*

Ce jeu d'attaque formalise l'idée suivante : $\mathcal{CS}$ est $IND\text{-}CPA$-sécurisé lorsqu'aucun adversaire à puissance polynomiale est capable de déterminer avec un avantage non négligeable lequel parmi deux messages de son choix a été choisi pour être chiffré. Nous en verrons d'autres caractérisations au chapitre suivant, dont une qui nous sera particulièrement utile pour étudier la confidentialité offerte par les protocoles d'authentification.

Le jeu ci-dessus est commode lorsqu'il s'agit d'analyser des protocoles utilisant un schéma de chiffrement. Par contre, le jeu équivalent suivant est plus facile à manier pour des arguments théoriques :

$$
\begin{aligned}
&\mathbf{Exp}^{ind-cpa}_{\mathcal{CS},\mathcal{A}}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (sk, pk) \leftarrow \mathcal{K}(I)\\
&\qquad (m_0, m_1, e) \leftarrow \mathcal{A}_1(I, pk)\\
&\qquad c_0, c_1 \leftarrow \mathcal{E}_{pk}(m_0), \mathcal{E}_{pk}(m_1)\\
&\qquad \gamma_0, \gamma_1 \leftarrow \mathcal{A}_2(m_0, m_1, e, c_0), \mathcal{A}_2(m_0, m_1, e, c_1)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}
$$

où on pose

$$Adv^{ind-cpa}_{\mathcal{CS},\mathcal{A}}(n) := \big|\mathbb{P}[\gamma_0 = 1] - \mathbb{P}[\gamma_1 = 1]\big|$$

Il est bien entendu supposé ci-dessus que les messages retournés par $\mathcal{A}_1$ ont même longueur.

Définition 9 *On dit que $\mathcal{CS}$ est ind-cpa-**sécurisé** si pour tout PPTA $\mathcal{A}$ comme ci-dessus l'application $Adv^{ind-cpa}_{\mathcal{CS},\mathcal{A}}$ est négligeable.*

Il est facile de voir que $\mathcal{CS}$ est *ind-cpa*-sécurisé si et seulement s'il est $IND\text{-}CPA$-sécurisé.

2.2.2 ElGamal et ElGamal bit-à-bit

Donnons maintenant un exemple concret de schéma de chiffrement. Il s'agit du schéma de chiffrement imaginé par ElGamal dans [10]. Pour pouvoir énoncer des résultats de sécurité bien connus à propos de ce schéma, nous devons d'abord définir la notion de famille de groupes et de générateur d'instance pour une telle famille.

Familles de groupes, hypothèse DDH

Définition 10 • *On appelle **famille de groupes** la donnée d'une famille $\mathfrak{G} := \{q_\lambda, \mathbb{G}_\lambda\}_\lambda$, où λ parcourt un ensemble infini, et pour tout λ, $\mathbb{G}_\lambda$ est un groupe cyclique d'ordre q_λ.*
• *Soit $\mathcal{G}$ un PPTA. On dit que $\mathcal{G}$ est un **générateur d'instance pour la famille de groupes** $\mathfrak{G}$ si pour tout $n \in \mathbb{N}$, $\mathcal{G}$ prend en entrée 1^n et retourne (la description d') un $(q_\lambda, \mathbb{G}_\lambda)$ de la famille $\mathfrak{G}$ et un générateur G de $\mathbb{G}_\lambda$.*

Soit $\mathfrak{G}$ une famille de groupes et $\mathcal{G}$ un générateur pour $\mathfrak{G}$. Les cryptosystèmes ElGamal et ElGamal bit-à-bit associés à $\mathcal{G}$ que nous allons définir plus loin ont une sécurité mesurée par la difficulté du problème décisionnel de Diffie-Hellman dans $\mathfrak{G}$. Précisément, il est bien connu que si ce problème est difficile dans $\mathfrak{G}$, le schéma de chiffrement ElGamal est $IND\text{-}CPA$-sécurisé. Nous montrerons dans la suite que si tel est bien le cas, le schéma de chiffrement ElGamal bit-à-bit est aussi $IND\text{-}CPA$-sécurisé.

La difficulté du problème décisionnel de Diffie-Hellman - abbrévié DDH dans la suite - peut aussi être quantifiée en termes de jeu d'attaque, où $\mathcal{A}$ est un PPTA : pour $n \in \mathbb{N}$, on considère

$$\begin{aligned}
&\mathbf{Exp}^{DDH}_{\mathcal{A},\mathcal{G}}(n)\\
&\quad (q_\lambda, \mathbb{G}_\lambda, G) \leftarrow \mathcal{G}(1^n)\\
&\quad\quad (x, y, z) \leftarrow \mathbb{Z}^3_{q_\lambda}\\
&\quad\quad\quad \gamma_0, \gamma_1 \leftarrow \mathcal{A}(q_\lambda, \mathbb{G}_\lambda, G, xG, yG, xyG), \mathcal{A}(q_\lambda, \mathbb{G}_\lambda, G, xG, yG, zG)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}$$

où le triplet (x, y, z) ci-dessus est choisi uniformément au hasard et

$$Adv^{DDH}_{\mathcal{G},\mathcal{A}}(n) := \left|\mathbb{P}[\gamma_0 = 1] - \mathbb{P}[\gamma_1 = 1]\right|$$

Définition 11 *Soit $\mathfrak{G}$ une famille de groupes et $\mathcal{G}$ un générateur d'instance pour $\mathfrak{G}$. On dit que $\mathcal{G}$ satisfait **l'hypothèse DDH** si pour tout PPTA $\mathcal{A}$ l'application $Adv^{DDH}_{\mathcal{G},\mathcal{A}}$ ci-dessus est négligeable.*

L'hypothèse DDH est une hypothèse très forte. Les premiers problèmes de calculs à apparaître dans la cryptographie utilisant des groupes cycliques, à savoir le problème du logarithme discret (DLP) et le problème calculatoire de Diffie-Hellman (DH), semblent beaucoup plus difficiles que ce problème décisionnel. Or, à l'heure actuelle, les seuls algorithmes connus pour résoudre DDH sont exactement ceux déjà utilisés pour résoudre DLP et DH. Pour un bon compte rendu du problème DDH, le lecteur pourra consulter [8].

ElGamal

Nous pouvons maintenant définir le schéma de chiffrement ElGamal associé à $\mathfrak{G}$, ou plus exactement au générateur d'instance $\mathcal{G}$ de $\mathfrak{G}$.

Construction du protocole

Définition 12 *Soit $\mathfrak{G}$ une famille de groupes et $\mathcal{G}$ un générateur d'instance pour $\mathfrak{G}$. Le **cryptosystème ElGamal $\mathcal{CS}^{EG}_{\mathcal{G}}$ associé à $\mathcal{G}$** est défini comme suit.*

- *Pour tout $n \in \mathbb{N}$, $\mathcal{P}$ prend en entrée 1^n, invoque $\mathcal{G}(1^n)$ pour obtenir $(q_\lambda, \mathbb{G}_\lambda, G)$ et retourne $(q_\lambda, \mathbb{G}_\lambda, G)$. L'espace des messages est $\mathbb{G}_\lambda$.*

• *Pour tout* $(q_\lambda, \mathbb{G}_\lambda, G)$, $\mathcal{K}$ *prend en entrée* (q_λ, G), *choisit* $x \in \mathbb{Z}_{q_\lambda}$ *uniformément au hasard, calcule* $Y := xG$ *et retourne* $(pk, sk) := (Y, x)$.

• *Pout tout* (Y, x) *et tout* $M \in \mathbb{G}_\lambda$, $\mathcal{E}$ *prend en entrée* (Y, M), *choisit* $r \in \mathbb{Z}_{q_\lambda}$ *uniformément au hasard, calcule* $U := rG$ *et* $V := M + rY$, *et retourne* (U, V). *On note* $\mathcal{E}_Y^{EG}$ *plutôt que* $\mathcal{E}_Y$.

• *Pour tout* (Y, x) *et tout* $(U, V) \in \mathcal{E}_Y^{EG}(M)$, $\mathcal{D}$ *prend en entrée* $\big(x, (U, V)\big)$, *calcule* $M' := V - xU$, *et retourne* M'. *On note* $\mathcal{D}_x^{EG}$ *plutôt que* $\mathcal{D}_x$.

Indistinguabilité des chiffrements

Pour montrer que le schéma de chiffrement que nous allons définir plus loin est également $IND\text{-}CPA$-sécurisé sous l'hypothèse DDH nous allons nous appuyer sur le théorème bien connu suivant :

Théorème 3 *Soient* $\mathfrak{G}$ *une famille de groupes et* $\mathcal{G}$ *un générateur d'instance pour* $\mathfrak{G}$. *Si* $\mathcal{G}$ *satisfait l'hypothèse DDH, le schéma de chiffrement ElGamal associé à* $\mathcal{G}$ *est* $IND\text{-}CPA$*-sécurisé.*

Voir [38] ou [34]. □

Le contenu concret de ce théorème s'exprime bien à l'aide d'un jeu d'attaque. Soient $\mathfrak{G}$, $\mathcal{G}$ et $\mathcal{CS}_{\mathcal{G}}^{EG}$ comme dans l'énoncé et $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ un PPTA opérant en deux temps. Le jeu est le suivant, où $n \in \mathbb{N}$:

$$
\begin{aligned}
&\mathbf{Exp}^{IND-CPA}_{\mathcal{CS}_{\mathcal{G}}^{EG},\mathcal{A}}(1^n)\\
&\qquad (q_\lambda, \mathbb{G}_\lambda, G) \leftarrow \mathcal{G}(1^n)\\
&\qquad (Y, x) \leftarrow \mathcal{K}(q_\lambda, \mathbb{G}_\lambda, G)\\
&\qquad (M_0, M_1, e) \leftarrow \mathcal{A}_1(q_\lambda, \mathbb{G}_\lambda, G, Y)\\
&\qquad \beta \leftarrow \{0, 1\}\\
&\qquad c \leftarrow \mathcal{E}_Y^{EG}(M_\beta)\\
&\qquad \gamma \leftarrow \mathcal{A}_2(M_0, M_1, e, c)\\
&\quad \text{Return } \gamma
\end{aligned}
$$

avec

$$Adv^{IND-CPA}_{\mathcal{CS}_{\mathcal{G}}^{EG},\mathcal{A}}(n) := \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|$$

Le théorème dit alors simplement que l'application $Adv^{IND-CPA}_{\mathcal{CS}_{\mathcal{G}}^{EG},\mathcal{A}}$ est négligeable.

ElGamal bit-à-bit

Définition du chiffrement

Définition 13 *Soit $\mathfrak{G}$ une famille de groupes et $\mathcal{G}$ un générateur d'instance pour $\mathfrak{G}$. Le* ***cryptosystème ElGamal bit-à-bit*** $\mathcal{CS}_{\mathcal{G}}^{EGbb}$ ***associé à*** $\mathcal{G}$ *est défini comme suit.*

• *Pour tout $n \in \mathbb{N}$, $\mathcal{P}$ prend en entrée 1^n, invoque $\mathcal{G}(1^n)$ pour obtenir $(q_\lambda, \mathbb{G}_\lambda, G)$ et retourne $(q_\lambda, \mathbb{G}_\lambda, G)$.*

• *Pour tout $(q_\lambda, \mathbb{G}_\lambda, G)$, $\mathcal{K}$ prend en entrée (q_λ, G), choisit $x \in \mathbb{Z}_{q_\lambda}$ uniformément au hasard, calcule $Y := xG$, choisit $S \in \mathbb{G}_\lambda$ non nul uniformément au hasard et retourne $(pk, sk) := \big((Y, S), x\big)$.*

• *Pour tout $\big((Y, S), x\big)$, tout $m \in \mathbb{N}$ et tout $M \in \{0,1\}^m$, $\mathcal{E}$ prend en entrée (Y, S), calcule pour tout $i \in \{0, ..., m-1\}$ un chiffrement ElGamal $c_i \leftarrow \mathcal{E}_Y^{EG}(M_i S)$ (où M_i désigne ici le $i-$ème bit de M) et retourne*

$$\mathcal{E}_{Y,S}(M) := (c_i)_{i \in \{0,...,m-1\}}$$

On note $\mathcal{E}_{Y,S}^{EGbb}$ plutôt que $\mathcal{E}_{Y,S}$.

• *Pour tout $\big((Y, S), x\big)$ et tout $(c_i)_i \in \mathcal{E}_{Y,S}^{EGbb}(M)$, $\mathcal{D}$ prend en entrée $(c_i)_i$ et x, calcule pour tout i*

$$m'_i = \mathcal{D}_{x,S}(c_i) = \begin{cases} 0 \ \ si \ \ \mathcal{D}_x^{EG}(c_i) = 0 \\ 1 \ \ si \ \ \mathcal{D}_x^{EG}(c_i) = S \\ 0 \ \ si \ \ \mathcal{D}_x^{EG}(c_i) = 2S \end{cases}$$

et retourne $(m'_i)_i$. On note $\mathcal{D}_x^{EGbb}$ plutôt que $\mathcal{D}_x$.

Homomorphie sous une addition

La formule de déchiffrement faisant intervenir le double du point S présent dans la clef publique nous permet de chiffrer de manière homomorphique sous une addition. C'est important pour le bon fonctionnement du protocole d'authentification construit à partir de ce chiffrement. Voyons comment marche cette propriété. L'addition est définie composante par composante, donc il nous suffit de regarder ce qui se passe au niveau d'un seul bit du vecteur.

Etant donnés deux bits b et b', on voudrait obtenir un chiffrement de $b \oplus b'$ en utilisant juste les éléments $\mathcal{E}_{Y,S}^{EGbb}(b)$ et $\mathcal{E}_{Y,S}^{EGbb}(b')$. En fait, on voudrait que la formule suivante soit vraie :

$$\mathcal{D}_x^{EGbb}\big(\mathcal{E}_{Y,S}^{EGbb}(b) + \mathcal{E}_{Y,S}^{EGbb}(b')\big) = b \oplus b'$$

Par abus de notation, cette équation est souvent écrite sous la forme

$$\mathcal{E}_{Y,S}^{EGbb}(b) + \mathcal{E}_{Y,S}^{EGbb}(b') = \mathcal{E}_{Y,S}^{EGbb}(b \oplus b') \quad (*)$$

Un calcul direct montre que pour le cryptosystème ElGamal on a pour deux éléments quelconques P et P' de $\mathbb{G}$ l'égalité

$$\mathcal{E}_Y^{EG}(P) + \mathcal{E}_Y^{EG}(P') = \mathcal{E}_Y^{EG}(P + P')$$

A l'aide de cette formule, on voit facilement que $(*)$ est vraie de manière naturelle si b et b' sont tous deux nuls ou distincts l'un de l'autre. Cependant, si $b = b' = 1$, ce qui est chiffré n'est plus 0 ou S mais bien le point $2S$. Pour contourner ce problème, nous décidons donc simplement que $2S$ correspond à 0, i.e.

$$\mathcal{D}_x^{EGbb}(2S) := 0$$

L'indistinguabilité des chiffrements ElGamal bit-à-bit

Pour étudier la sécurité de ce schéma de chiffrement, il faut prendre un peu de précautions car il n'est pas évident à priori que l'espèce d'homomorphie forcée que nous utilisons ne présente pas un danger. Pour montrer que ce n'est pas le cas, nous allons d'abord montrer que le schéma est $IND\text{-}CPA$-sécurisé sans nous préoccuper de la formule d'homomorphie, et ensuite montrer comment cela implique que des chiffrements obtenus par addition ne sont pas dangereux.

Indistinguabilité simple

Soient maintenant $\mathfrak{G} := \{q_\lambda, \mathbb{G}_\lambda\}_\lambda$ une famille de groupes et $\mathcal{G}$ un générateur d'instance pour $\mathfrak{G}$. Notons $\mathcal{CS}^{EG}$ et $\mathcal{CS}^{EGbb}$ respectivement les schémas de chiffrement ElGamal et ElGamal bit-à-bit associés à $\mathcal{G}$. Nous allons montrer la

Proposition 1 *Si $\mathcal{G}$ satisfait l'hypothèse DDH, $\mathcal{CS}^{EGbb}$ est ind-cpa-sécurisé.*

La preuve se fait en plusieurs étapes. D'abord, il est bien connu que l'hypothèse de l'énoncé entraîne que $\mathcal{CS}^{EG}$ est *ind-cpa*-sécurisé. Il suffit donc de montrer la

Proposition 2 *Si $\mathcal{CS}^{EG}$ est ind-cpa-sécurisé, $\mathcal{CS}^{EGbb}$ l'est également.*

Ensuite, la proposition se déduit des deux lemmes suivants, où $\mathcal{CS}_1^{EGbb}$ désigne le schéma de chiffrement ElGamal bit-à-bit où l'on suppose que les messages clairs ne peuvent être que de longueur 1, i.e. n'est qu'un seul bit :

Lemme 1 *Si $\mathcal{CS}^{EG}$ est ind-cpa-sécurisé, $\mathcal{CS}_1^{EGbb}$ l'est également.*

et

Lemme 2 *Si $\mathcal{CS}_1^{EGbb}$ est ind-cpa-sécurisé, $\mathcal{CS}^{EGbb}$ est ind-cpa-sécurisé.*

Avant de faire les démonstrations, nous allons expliciter les jeux d'attaque $\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}^{EG},\mathcal{A}}$, $\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}_1^{EGbb},\mathcal{A}}$ et $\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}^{EGbb},\mathcal{A}}$. Les avantages correspondants s'en déduisent sans peine.

Soit $n \in \mathbb{N}$ le paramètre de sécurité.

Le jeu contre ElGamal :

$$
\begin{aligned}
&\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}^{EG},\mathcal{A}}(n)\\
&\quad (q_\lambda, \mathbb{G}_\lambda, G) \leftarrow \mathcal{G}(1^n)\\
&\quad x \leftarrow \mathbb{Z}_{q_\lambda}\\
&\quad Y \leftarrow xG\\
&\quad (M_0, M_1, e) \leftarrow \mathcal{A}_1(q_\lambda, \mathbb{G}_\lambda, G, Y)\\
&\quad r_0, r_1 \leftarrow \mathbb{Z}_{q_\lambda}, \mathbb{Z}_{q_\lambda}\\
&\quad U_0, U_1 \leftarrow r_0 G, r_1 G\\
&\quad T_0, T_1 \leftarrow r_0 Y, r_1 Y\\
&\quad V_0, V_1 \leftarrow M_0 + T_0, M_1 + T_1\\
&\quad \gamma_0, \gamma_1 \leftarrow \mathcal{A}_2\big(M_0, M_1, e, (U_0, V_0)\big), \mathcal{A}_2\big(M_0, M_1, e, (U_1, V_1)\big)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}
$$

Le jeu contre ElGamal bit-à-bit avec 1 bit :

$$
\begin{aligned}
&\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}_1^{EGbb},\mathcal{A}}(s)\\
&\quad (q_\lambda, \mathbb{G}_\lambda, G) \leftarrow \mathcal{G}(1^s)\\
&\quad x \leftarrow \mathbb{Z}_{q_\lambda}\\
&\quad Y \leftarrow xG\\
&\quad S \leftarrow \mathbb{G}_\lambda\\
&\quad 0, 1 \leftarrow \mathcal{A}_1(q_\lambda, \mathbb{G}_\lambda, G, Y, S)\\
&\quad r_0, r_1 \leftarrow \mathbb{Z}_{q_\lambda}, \mathbb{Z}_{q_\lambda}\\
&\quad U_0, U_1 \leftarrow r_0 G, r_1 G\\
&\quad T_0, T_1 \leftarrow r_0 Y, r_1 Y\\
&\quad V_0, V_1 \leftarrow T_0, S + T_1\\
&\quad \gamma_0, \gamma_1 \leftarrow \mathcal{A}_2\big(0, 1, e, (U_0, V_0)\big), \mathcal{A}_2\big(0, 1, e, (U_1, V_1)\big)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}
$$

Le jeu contre ElGamal bit-à-bit à longueur arbitraire :

$$
\begin{aligned}
&\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}^{EGbb},\mathcal{A}}(s)\\
&\quad (q_\lambda, \mathbb{G}_\lambda, G) \leftarrow \mathcal{G}(1^s)\\
&\quad x \leftarrow \mathbb{Z}_{q_\lambda}\\
&\quad Y \leftarrow xG\\
&\quad S \leftarrow \mathbb{G}_\lambda\\
&\quad m_0, m_1 \leftarrow \mathcal{A}_1(q_\lambda, \mathbb{G}_\lambda, G, Y, S)\\
&\quad (r_{0i})_i, (r_{1i})_i \leftarrow (\mathbb{Z}_{q_\lambda})_i, (\mathbb{Z}_{q_\lambda})_i\\
&\quad (U_{0i})_i, (U_{1i})_i \leftarrow (r_{0i}G)_i, (r_{1i}G)_i\\
&\quad (T_{0i})_i, (T_{1i})_i \leftarrow (r_{0i}Y)_i, (r_{1i}Y)_i\\
&\quad (V_{0i})_i, (V_{1i})_i \leftarrow (m_{0i}S + T_{0i})_i, (m_{1i}S + T_{1i})_i\\
&\quad \gamma_0, \gamma_1 \leftarrow \mathcal{A}_2\big(m_0, m_1, e, (U_{0i}, V_{0i})_i\big), \mathcal{A}_2\big(m_0, m_1, e, (U_{1i}, V_{1i})_i\big)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}
$$

On commence par le lemme 1. C'est assez simple : si un attaquant peut distinguer les chiffrements de 0 des chiffrements de 1, il peut distinguer les chiffrements de l'élément neutre de $\mathbb{G}_\lambda$ des chiffrements d'autres points de $\mathbb{G}_\lambda$ choisis uniformément au hasard.

Preuve du lemme 1

Soit $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$ un PPTA jouant contre $\mathcal{CS}_1^{EGbb}$ dans le jeu $\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}_1^{EGbb},\mathcal{B}}$ et construisons un PPTA $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ qui joue contre $\mathcal{CS}^{EG}$ dans le jeu $\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}^{EG},\mathcal{A}}$ en utilisant $\mathcal{B}$ comme sous-programme. Soit $n \in \mathbb{N}$.

- Le challengeur $\mathcal{CH}$ invoque $\mathcal{G}(1^n)$ pour obtenir $(q_\lambda, \mathbb{G}_\lambda, G)$ puis choisit $x \in \mathbb{Z}_{q_\lambda}$ uniformément au hasard et pose $Y = xG$. Il envoie alors $(q_\lambda, \mathbb{G}_\lambda, G, Y)$ à $\mathcal{A}_1$.
- Le PPTA $\mathcal{A}_1$ choisit $S \in \mathbb{G}_\lambda$ uniformément au hasard, pose $M_0 = 0$ et $M_1 = S$, et invoque $\mathcal{B}_1(q_\lambda, \mathbb{G}_\lambda, G, Y, S)$ pour obtenir $(0, 1, e_\mathcal{B})$. Ensuite, $\mathcal{A}_1$ pose $e_\mathcal{A} = e_\mathcal{B}$ et envoie (M_0, M_1) à $\mathcal{CH}$ et $(M_0, M_1, e_\mathcal{A})$ à $\mathcal{A}_2$.
- Le challengeur choisit $\beta \in \{0, 1\}$ uniformément au hasard, choisit $r \in \mathbb{Z}_{q_\lambda}$ uniformément au hasard, calcule $U := rG$, $T := rY$ et $V := M_\beta + T$, et envoie (U, V) à $\mathcal{A}_2$.
- Le PPTA $\mathcal{A}_2$ invoque $\mathcal{B}_2\big(0, 1, e_\mathcal{B}, (U, V)\big)$ pour obtenir γ et retourne 0 si $\gamma = 0$ et S si $\gamma = 1$.

Passons à l'analyse des avantages. D'abord, le choix de S fait par $\mathcal{A}_1$ respecte la création de la clef publique de $\mathcal{CS}_1^{EGbb}$. Ensuite, on voit tout de suite que $\mathcal{A}_2$ retourne M_β si et seulement si $\mathcal{B}_2$ retourne β. Il en résulte finalement que $Adv^{ind-cpa}_{\mathcal{CS}_1^{EGbb},\mathcal{A}} = Adv^{ind-cpa}_{\mathcal{CS}^{EG},\mathcal{B}}$ d'où le lemme. □

Passons à la preuve du lemme 2. Intuitivement, pour passer d'un bit à des messages de longueur k plus grande, il faut remarquer que si on peut distinguer entre les chiffrements de deux messages de longueur k distincts, on peut exhiber deux messages de longueur k, distincts l'un de l'autre en une seule position et dont les chiffrements sont distingables. On utilise alors le challenge pour chiffrer ce bit et en déduire la réponse.

Preuve du lemme 2

Ce lemme utilise un argument hybride. Soit $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$ un PPTA jouant contre $\mathcal{CS}^{EGbb}$ selon $\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}^{EGbb},\mathcal{B}}$ et construisons un PPTA $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ jouant contre $\mathcal{CS}_1^{EGbb}$ en invoquant $\mathcal{B}$. Soit n le paramètre de sécurité.

• Le challengeur $\mathcal{CH}$ invoque $\mathcal{G}(1^n)$ pour obtenir $(q_\lambda, \mathbb{G}_\lambda, G)$, choisit $x \in \mathbb{Z}_{q_\lambda}$ uniformément au hasard, pose $Y = xG$ et choisit $S \in \mathbb{G}_\lambda$ uniformément au hasard. Il envoie ensuite $(q_\lambda, \mathbb{G}_\lambda, G, Y, S)$ à $\mathcal{A}_1$.

• L'attaquant $\mathcal{A}_1$ invoque $\mathcal{B}_1(q_\lambda, \mathbb{G}_\lambda, G, Y, S)$ pour obtenir m_0 et m_1 de même longueur dans $\{0,1\}^*$ et l'information d'état $e_\mathcal{B}$. Notons k la longueur de m_0 et m_1. Comme $\mathcal{B}$ opère en temps polynomial, k est polynomial en n. Puis, $\mathcal{A}$ pose $e_\mathcal{A} = (m_0, m_1, e_\mathcal{B})$, envoie $(0, 1, e_\mathcal{A})$ à $\mathcal{A}_2$ et $(0,1)$ à $\mathcal{CH}$.

• Le challengeur choisit $\beta \in \{0,1\}$ uniformément au hasard, choisit $r \in \mathbb{Z}_{q_\lambda}$ uniformément au hasard, calcule $U := rG$, $T := rY$ et $V := \beta S + T$ et envoie (U, V) à $\mathcal{A}_2$.

• Le PPTA $\mathcal{A}_2$ extrait m_0 et m_1 de $e_\mathcal{A}$ et exploite les différences entre ces deux messages. Soit μ le poids de Hamming de $m_0 \oplus m_1$; $\mathcal{A}_2$ calcule une chaîne de longueur μ allant de m_0 à m_1 :

$$m_0 =: m^{(0)}, m^{(1)}, ..., m^{(\mu)} := m_1$$

Pour tout $j \in \{0, ..., \mu-1\}$, on note h_j la composante où $m^{(j)}$ et $m^{(j+1)}$ diffèrent. L'adversaire $\mathcal{A}_2$ choisit $a \in \{0, ..., \mu - 1\}$ uniformément au hasard et travaille avec $m^{(a)}$ et $m^{(a+1)}$.

Pour tout $h \in \{0, ..., k-1\}$ distinct de h_a, $\mathcal{A}_2$ choisit $r_h \in \mathbb{Z}_{q_\lambda}$ uniformément au hasard et calcule $U_h := r_h G$, $T_h := r_h Y$ et $V_h := m_h^{(a)} S + T_h$. On remarque que $m_h^{(a)} = m_h^{(a+1)}$ vu que $h \neq h_a$. Puis, $\mathcal{A}_2$ pose $U_{h_a} = U$ et $V_{h_a} = V$.

Ensuite $\mathcal{A}_2$ extrait $e_\mathcal{B}$ de $e_\mathcal{A}$ et invoque $\mathcal{B}_2\big(m^{(a)}, m^{(a+1)}, e_\mathcal{B}, (U_h, V_h)_h\big)$ pour obtenir γ. Enfin, $\mathcal{A}_2$ retourne γ si $m_{h_a}^{(a)} = 0$ et $\gamma \oplus 1$ si $m_{h_a}^{(a+1)} = 1$.

Analysons l'avantage de $\mathcal{A}$ en fonction de celui de $\mathcal{B}$. D'abord, il est clair que les clefs passées à $\mathcal{B}$ sont générées selon les spécifications de $\mathcal{CS}^{EGbb}$. Ensuite, examinons le vecteur $(U_h, V_h)_h$. Pour tout $h \neq h_a$, (U_h, V_h) est un chiffrement de $m_h^{(a)} = m_h^{(a+1)}$ donc $(U_h, V_h)_h$ est un chiffrement de $m^{(a)}$ ou $m^{(a+1)}$ selon

• la valeur choisie par $\mathcal{CH}$ que $\mathcal{A}$ essaye de déterminer et

• la valeur de $m_{h_a}^{(a)}$ qui par construction est distincte de celle de $m^{(a+1)}$ et que $\mathcal{A}$ connaît.

Pour expliciter les avantages des attaquants, il est commode d'adopter la notation $\mathcal{E}_{Y,S}^{EGbb}$ pour désigner des chiffrements de messages. On a d'une part :

$$\begin{aligned}
&\mathbb{P}\Big[\mathcal{A}_2\big(0,1,e_{\mathcal{A}},\mathcal{E}_{Y,S}^{EGbb}(0)\big)=1\Big]= \\
&\quad\sum_{j,m_{h_j}^{(j)}=0}\mathbb{P}\Big[\mathcal{B}_2\big(m^{(a)},m^{(a+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(a)})\big)=1\Big|a=j\Big]\mathbb{P}[a=j]+ \\
&\quad\sum_{j,m_{h_j}^{(j)}=1}\mathbb{P}\Big[\mathcal{B}_2\big(m^{(a)},m^{(a+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}(m^{(a+1)})\big)=0\Big|a=j\Big]\mathbb{P}[a=j]= \\
&\qquad\frac{1}{\mu}\Big(\sum_{j,m_{h_j}^{(j)}=0}\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j)})\big)=1\Big]+ \\
&\qquad\qquad\sum_{j,m_{h_j}^{(j)}=1}\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}(m^{(j+1)})\big)=0\Big]\Big)
\end{aligned}$$

et (de la même manière)

$$\begin{aligned}
&\mathbb{P}\Big[\mathcal{A}_2\big(0,1,e_{\mathcal{A}},\mathcal{E}_{pk}(1)\big)=1\Big]= \\
&\qquad\frac{1}{\mu}\Big(\sum_{j,m_{h_j}^{(j)}=0}\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j+1)})\big)=1\Big]+ \\
&\qquad\qquad\sum_{j,m_{h_j}^{(j)}=1}\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EG}(m^{(j)})\big)=0\Big]\Big)
\end{aligned}$$

d'où

$$\begin{aligned}
&\Big|\mathbb{P}\Big[\mathcal{A}_2\big(0,1,e_{\mathcal{A}},\mathcal{E}_{Y,S}^{EGbb}(0)\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(0,1,e_{\mathcal{A}},\mathcal{E}_{Y,S}^{EGbb}(1)\big)=1\Big]\Big|= \\
&\qquad\frac{1}{\mu}\Big|\sum_{j,m_{h_j}^{(j)}=0}\Big(\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j)})\big)=1\Big]- \\
&\qquad\qquad\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j+1)})\big)=1\Big]\Big)+ \\
&\qquad\sum_{j,m_{h_j}^{(j)}=1}\Big(\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j+1)})\big)=0\Big]- \\
&\qquad\qquad\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j)})\big)=0\Big]\Big)\Big|=
\end{aligned}$$

$$\frac{1}{\mu}\Bigg|\sum_{j,m_{h_j}^{(j)}=0}\Big(\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j)})\big)=1\Big]-$$
$$\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}(m^{(j+1)})\big)=1\Big]\Big)+$$
$$\sum_{j,m_{h_j}^{(j)}=1}\Big(\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j)})\big)=1\Big]-$$
$$\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j+1)})\big)=1\Big]\Big)\Bigg|$$

et d'autre part,

$$\Big|\mathbb{P}\Big[\mathcal{B}_2\big(m_0,m_1,e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m_0)\big)=1\Big]-\mathbb{P}\Big[\mathcal{B}_2\big(m_0,m_1,e,\mathcal{E}_{Y,S}^{EGbb}(m_1)\big)=1\Big]\Big|=$$
$$\Bigg|\sum_{j=0}^{\mu-1}\mathbb{P}\Big(\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j)})\big)=1\Big]-$$
$$\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j+1)})\big)=1\Big]\Big)\Bigg|=$$
$$\Bigg|\sum_{j,m_{h_j}^{(j)}=0}\Big(\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j)})\big)=1\Big]-$$
$$\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j+1)})\big)=1\Big]\Big)+$$
$$\sum_{j,m_{h_j}^{(j)}=1}\Big(\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j)})\big)=1\Big]-$$
$$\mathbb{P}\Big[\mathcal{B}_2\big(m^{(j)},m^{(j+1)},e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m^{(j+1)})\big)=1\Big]\Big)\Bigg|$$

donc on a au final

$$\Big|\mathbb{P}\Big[\mathcal{A}_2\big(0,1,e_{\mathcal{A}},\mathcal{E}_{Y,S}^{EGbb}(0)\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(0,1,e_{\mathcal{A}},\mathcal{E}_{Y,S}^{EGbb}(1)\big)=1\Big]\Big|\geq$$
$$\frac{1}{k}\Big|\mathbb{P}\Big[\mathcal{B}_2\big(m_0,m_1,e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m_0)\big)=1\Big]-\mathbb{P}\Big[\mathcal{B}_2\big(m_0,m_1,e_{\mathcal{B}},\mathcal{E}_{Y,S}^{EGbb}(m_1)\big)=1\Big]\Big|$$

et il en résulte

$$Adv_{\mathcal{CS}_1^{EGbb},\mathcal{A}}^{ind-cpa}\geq\frac{1}{k}Adv_{\mathcal{CS}^{EGbb},\mathcal{B}}^{ind-cpa}$$

La négligeabilité de $Adv_{\mathcal{CS}_1^{EGbb},\mathcal{A}}^{ind-cpa}$ entraîne donc celle de $Adv_{\mathcal{CS}^{EGbb},\mathcal{B}}^{ind-cpa}$ car k est borné par le temps de calcul de $\mathcal{B}$ qui est un PPTA. Ceci achève la preuve. □

L'effet de l'homomorphie

Pour étudier l'effet de la formule d'homomorphie sur la sécurité, nous allons introduire un autre jeu d'attaque. L'intuition derrière ce jeu est directement lisible dans le jeu lui-même : on demande à un adversaire de choisir deux messages et deux masques et il doit distinguer la somme du chiffrement d'un message avec celui d'un masque de la somme du chiffrement de l'autre message avec l'autre masque. On remarque que si les masques choisis sont tous deux identiquement nuls on tombe juste sur une manière un peu compliquée de formuler l'*ind-cpa*-sécurité simple.

Soient $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ un PPTA et n le paramètre de sécurité.

$$
\begin{aligned}
&\mathbf{Exp}^{ind-cpa-h}_{\mathcal{CS}^{EGbb},\mathcal{A}}(n)\\
&\qquad (q_\lambda, \mathbb{G}_\lambda, G) \leftarrow \mathcal{G}(1^n)\\
&\qquad x \leftarrow \mathbb{Z}_{q_\lambda}\\
&\qquad Y \leftarrow xG\\
&\qquad S \leftarrow \mathbb{G}_\lambda\\
&\qquad (m_0, m'_0, m_1, m'_1, e) \leftarrow \mathcal{A}_1(q_\lambda, \mathbb{G}_\lambda, G, Y, S)\\
&\qquad (r_{0i})_i, (r_{1i})_i, (r'_{0i})_i, (r'_{1i})_i \leftarrow \mathbb{Z}^k_{q_\lambda}, \mathbb{Z}^k_{q_\lambda}, \mathbb{Z}^k_{q_\lambda}, \mathbb{Z}^k_{q_\lambda}\\
&\qquad (U_{0i})_i, (U_{1i})_i, (U'_{0i})_i, (U'_{1i})_i \leftarrow (r_{0i}G)_i, (r_{1i}G)_i, (r'_{0i}G)_i, (r'_{1i}G)_i\\
&\qquad (T_{0i})_i, (T_{1i})_i, (T'_{0i})_i, (T'_{1i})_i \leftarrow (r_{0i}Y)_i, (r_{1i}Y)_i, (r'_{0i}Y)_i, (r'_{1i}Y)_i\\
&\qquad (V_{0i})_i, (V_{1i})_i \leftarrow (m_{0i}S + T_{0i})_i, (m_{1i}S + T_{1i})_i\\
&\qquad (V'_{0i})_i, (V'_{1i})_i \leftarrow (m'_{0i}S + T'_{0i})_i, (m'_{1i}S + T'_{1i})_i\\
&\qquad \gamma_0 \leftarrow \mathcal{A}_2\big(m_0, m'_0, m_1, m'_1, e, (U_{0i} + U'_{0i}, V_{0i} + V'_{0i})_i\big)\\
&\qquad \gamma_1 \leftarrow \mathcal{A}_2\big(m_0, m'_0, m_1, m'_1, e, (U_{1i} + U'_{1i}, V_{1i} + V'_{1i})_i\big)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}
$$

L'entier k est la longueur des messages (et des masques) choisis par $\mathcal{A}$ et l'avantage correspondant est

$$Adv^{ind-cpa-h}_{\mathcal{CS}^{EGbb},\mathcal{A}} := \big|\mathbb{P}[\gamma_0 = 1] - \mathbb{P}[\gamma_1 = 1]\big|$$

La définition suivante est très locale :

Définition 14 *On dit que $\mathcal{CS}^{EGbb}$ a des **chiffrements homomorphiques indistinguables sous une attaque à messages clairs choisis** (ou est ind-cpa-h-sécurisé) si pour tout PPTA $\mathcal{A}$ l'application $Adv^{ind-cpa-h}_{\mathcal{CS}^{EGbb},\mathcal{A}}$ est négligeable.*

Maintenant, on a le résultat suivant :

Proposition 3 *Si $\mathcal{CS}^{EGbb}$ est ind-cpa-sécurisé, $\mathcal{CS}^{EGbb}$ est ind-cpa-h-sécurisé.*

Soit $\mathcal{B}$ un PPTA jouant contre $\mathcal{CS}^{EGbb}$ dans le jeu $\mathbf{Exp}^{ind-cpa-h}_{\mathcal{CS}^{EGbb},\mathcal{B}}$. On construit un PPTA $\mathcal{A}$ jouant contre $\mathcal{CS}^{EGbb}$ dans le jeu $\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}^{EGbb},\mathcal{A}}$ utilisant $\mathcal{B}$ comme sous-fiffre.

• Le challengeur $\mathcal{CH}$ invoque $\mathcal{G}(1^n)$ pour obtenir $(q_\lambda, \mathbb{G}_\lambda, G)$, choisit $x \in \mathbb{Z}_{q_\lambda}$ et $S \in \mathbb{G}_\lambda$ uniformément au hasard, calcule $Y := xG$ et envoie $(q_\lambda, \mathbb{G}_\lambda, G, Y, S)$ à $\mathcal{A}_1$.

• Le PPTA $\mathcal{A}_1$ invoque $\mathcal{B}_1(q_\lambda, \mathbb{G}_\lambda, G, Y, S)$ pour obtenir $(m_0, m_1, m'_0, m'_1, e_\mathcal{B})$. Il pose $e_\mathcal{A} = e_\mathcal{B}$, $m''_0 = m_0||m'_0$ et $m''_1 = m_1||m'_1$, envoie $(m''_0, m''_1, e_\mathcal{A})$ à $\mathcal{A}_2$ et (m''_0, m''_1) à $\mathcal{CH}$. Soit k la longueur commune des messages émis.

• Le challengeur $\mathcal{CH}$ choisit $\beta \in \{0,1\}$ uniformément au hasard et choisit $(r''_i)_i$ uniformément au hasard pour i parcourant $\{0, ..., 2k-1\}$. Ensuite, il calcule $(U''_i)_i := (r''_i G)_i$, $(T''_i)_i := (r''_i Y)_i$ et $(V''_i)_i := (m''_{\beta i} S + T''_i)_i$, et envoie $(U''_i, V''_i)_i$ à $\mathcal{A}_2$.

• Pour tout $i \in \{0, ..., k-1\}$, l'attaquant $\mathcal{A}_2$ calcule alors

$$(U'''_i, V'''_i) := (U''_i, V''_i) + (U''_{k+i}, V''_{k+i}),$$

invoque $\mathcal{B}_2\big(m_0, m'_0, m_1, m'_1, e_\mathcal{B}, (U'''_i, V'''_i)_{i\in\{0,...,k-1\}}\big)$ pour obtenir γ, et retourne γ.

L'*ind-cpa*-sécurité de $\mathcal{CS}^{EGbb}$ est utilisée de manière un peu surprenante ici : $\mathcal{A}$ a besoin de chiffrer l'un des deux masques pour ensuite le sommer au chiffrement qu'il reçoit en challenge. Or, il ne sait pas quel message est choisi par $\mathcal{CH}$ donc il ne peut décider quel masque chiffrer. Il demande donc à $\mathcal{CH}$ de faire ce choix, en "cachant" le masque en deuxième partie d'un message deux fois plus grand. Ce n'est pas un incovénient car la longueur des messages résultant de cette opération reste polynomiale en n.

Il reste à analyser le rapport entre les avantages. Il est clair que les chiffrements obtenus de $\mathcal{CH}$ et sommés par $\mathcal{A}_2$ respectent les spécifications de $\mathbf{Exp}^{ind-cpa-h}_{\mathcal{CS}^{EGbb},\mathcal{B}}$ et on voit que $\mathcal{A}_2$ répond correctement si et seulement s'il en est de même de $\mathcal{B}_2$. On a donc

$$Adv^{ind-cpa-h}_{\mathcal{CS}^{EGbb},\mathcal{B}} = Adv^{ind-cpa}_{\mathcal{CS}^{EGbb},\mathcal{A}}$$

et la négligeabilité du membre de droite entraîne celle du membre de gauche. □

On peut désormais conclure ce paragraphe avec le

Corollaire 1 *Si $\mathcal{CS}^{EGbb}_1$ est ind-cpa-sécurisé, $\mathcal{CS}^{EGbb}$ est ind-cpa-sécurisé et ind-cpa-h-sécurisé.*

En effet, que $\mathcal{CS}^{EGbb}_1$ soit *ind-cpa*-sécurisé entraîne que $\mathcal{CS}^{EGbb}$ soit *ind-cpa*-sécurisé. Il reste alors à appliquer la proposition 3. □

Résultat global

On peut résumer les résultats obtenus dans les paragraphes précédents dans un théorème.

Théorème 4 *Soient $\mathfrak{G}$ une famille de groupes, $\mathcal{G}$ un générateur d'instance pour $\mathfrak{G}$. Si $\mathcal{G}$ satisfait l'hypothèse DDH, le schéma de chiffrement $\mathcal{CS}^{EGbb}$ est ind-cpa-sécurisé et ind-cpa-h-sécurisé.*

2.3 Signatures digitales et fonctions de hachage

Ici nous faisons des rappels sur les schémas de signature digitale et les fonctions de hachage et certains aspects de leur sécurité. Nous concluons en définissant un jeu d'attaque contre un schéma de signature et une fonction de hachage en même temps afin d'avoir un formalisme bien adapté au modèle de sécurité. Ces primitives sont utilisées pour garantir l'intégrité de la réponse finale d'une authentification. Il est clair que toute la machinerie que l'on souhaite mettre en place ne sert à rien si un adversaire a la possibilité de changer cette réponse.

2.3.1 Signatures

Définition 15 *Un **schéma de signature digitale** est un quadruplet*

$$\mathcal{DS} := (\mathcal{P}_s, \mathcal{K}_s, Sign, Verify)$$

de PPTA vérifiant les hypothèses suivantes :

- *Pour tout $n \in \mathbb{N}$, $\mathcal{P}_s$ prend en entrée 1^n et retourne un ensemble de paramètres I_s. Ces paramètres spécifient notamment un espace de messages $\mathcal{M}_s$ à signer et le paramètre de sécurité n de départ.*
- *Pour tout $I_s \in \mathcal{P}_s(1^n)$, $\mathcal{K}_s$ prend en entrée I_s et retourne une paire de clefs (pk_s, sk_s). On appelle pk_s et sk_s les clefs de vérification et de signature respectivement.*
- *Pour tout $(pk_s, sk_s) \in \mathcal{K}_s(I_s)$, et tout $m \in \mathcal{M}_s$, Sign prend en entrée sk_s et m et retourne $\mathfrak{s}$. On note $Sign_{sk_s}(m)$ plutôt que $Sign(sk_s, m)$.*
- *Pour tout $\mathfrak{s} \in Sign_{sk_s}(m)$ et tout $m' \in \mathcal{M}_s$, $Verify$ prend en entrée pk_s, $\mathfrak{s}$ et m' et retourne 0 ou 1. On note $Verify_{pk_s}(m', \mathfrak{s})$ plutôt que $Verify(pk_s, m', \mathfrak{s})$.*
- *Pour tout $m \in \mathcal{M}_s$ on a*

$$Verify_{pk_s}\big(m, Sign_{sk_s}(m)\big) = 1$$

L'attaquant naturel contre une signature digitale est un PPTA cherchant à trouver des signatures forgées à partir uniquement de la clef publique. Soient $n \in \mathbb{N}$, $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ un PPTA fonctionnant en deux étapes et $\mathcal{OR}_{Sign_{sk_s}}$ un oracle de signature. Considérons le jeu suivant :

$$\begin{aligned}
&\mathbf{Exp}^{for}_{\mathcal{DS},\mathcal{A}}(n) \\
&\qquad I_s \leftarrow \mathcal{P}_s(1^n) \\
&\qquad (pk_s, sk_s) \leftarrow \mathcal{K}_s(I_s) \\
&\qquad \big((m_o, \mathfrak{s}_o)_o, e\big) \leftarrow \mathcal{A}_1(I_s, pk_s, \mathcal{OR}_{Sign_{sk_s}}) \\
&\qquad (m, \mathfrak{s}) \leftarrow \mathcal{A}_2\big((m_o, \mathfrak{s}_o)_o, e\big) \\
&\text{Return } (m, \mathfrak{s})
\end{aligned}$$

où on définit l'avantage de $\mathcal{A}$ comme étant

$$Adv^{for}_{\mathcal{DS},\mathcal{A}}(n) := \mathbb{P}\left[\Big((m, \mathfrak{s}) \notin \big\{(m_o, \mathfrak{s}_o)\big\}_o\Big) \wedge \big(Verify_{pk_s}(m, \mathfrak{s}) = 1\big)\right]$$

Définition 16 *On dit que $\mathcal{DS}$ a des **signatures inforgeables** si pour tout PPTA $\mathcal{A}$ l'application $Adv^{for}_{\mathcal{DS},\mathcal{A}}$ est négligeable en le paramètre de sécurité.*

2.3.2 Fonctions de hachage

Définition 17 *Soient $r : \mathbb{N} \to \mathbb{N}$ une application et $\mathfrak{H} := \{H_\lambda\}_\lambda$ une famille d'applications $H_\lambda : \{0,1\}^* \to \{0,1\}^{r(|\lambda|)}$ où les λ parcourent un sous-ensemble infini de $\{0,1\}^*$. On dit que $\mathfrak{H}$ est une **famille de fonctions de hachage** si les conditions suivantes sont vérifiées :*

- *il existe un PPTA $\mathcal{GH}$ tel que pour tout $n \in \mathbb{N}$, $\mathcal{GH}$ prenne en entrée 1^n et retourne un λ et la description de H_λ et*
- *pour tout λ l'application H_λ est calculable en temps polynomial.*

Soient $\mathcal{A}$ un PPTA et $n \in \mathbb{N}$ le paramètre de sécurité. On considère le jeu suivant :

$$\begin{aligned} &\mathbf{Exp}^{coll}_{\mathfrak{H},\mathcal{A}}(n) \\ &\quad \big(\lambda, \langle H_\lambda \rangle\big) \leftarrow \mathcal{GH}(1^n) \\ &\quad (x_0, x_1) \leftarrow \mathcal{A}\big(\lambda, \langle H_y \rangle\big) \\ &\text{Return } (x_0, x_1) \end{aligned}$$

où on définit l'avantage de $\mathcal{A}$ par

$$Adv^{coll}_{\mathfrak{H},\mathcal{A}}(n) := \mathbb{P}\Big[(x_0 \neq x_1) \wedge \big(H_\lambda(x_0) = H_\lambda(x_1)\big)\Big]$$

Définition 18 *On dit que la famille de fonctions de hachage $\mathfrak{H}$ **résiste aux collisions** si pour tout PPTA $\mathcal{A}$ l'application $Adv^{coll}_{\mathfrak{H},\mathcal{A}}$ est négligeable.*

On remarque que pour qu'une famille de fonctions de hachage résiste aux collisions il faut que r croisse suffisamment de sorte à ce qu'une recherche exhaustive soit impossible en temps polynomial.

2.3.3 Un jeu d'attaque simultané

Soient $\mathcal{DS}$ un schéma de signature et $\mathfrak{H}$ une famille de fonctions de hachage. Pour tout PPTA $\mathcal{A}$, on définit le jeu d'attaque suivant pour $n \in \mathbb{N}$:

$$\begin{aligned} &\mathbf{Exp}^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}}(n) \\ &\quad (I_s) \leftarrow \mathcal{P}_s(1^n) \\ &\quad (pk_s, sk_s) \leftarrow \mathcal{K}_s(I) \\ &\quad \big(\lambda, \langle H_\lambda \rangle\big) \leftarrow \mathcal{GH}(1^n) \\ &\quad \big((m_o, \mathfrak{s}_o)_o, e\big) \leftarrow \mathcal{A}_1\big(pk_s, \langle H_\lambda \rangle, \mathcal{OR}_{Sign_{sk_s}}\big) \\ &\quad (\gamma, \sigma) \leftarrow \mathcal{A}_2\big((m_o, \mathfrak{s}_o)_o, e\big) \\ &\text{Return } (\gamma, \sigma) \end{aligned}$$

où l'avantage est défini comme étant

$$Adv^{coll/for}_{\mathfrak{H},\mathcal{SD},\mathcal{A}}(n) := \mathbb{P}\Bigg[\Big((\gamma \neq \sigma) \wedge \big(H_\lambda(\gamma) = H_\lambda(\sigma)\big)\Big) \vee \Big(\Big((\gamma,\sigma) \notin \big\{(m_o,\mathfrak{s}_o)\big\}_o\Big) \wedge \big(Verify_{pk_s}(\gamma,\sigma) = 1\big)\Big)\Bigg]$$

Ce jeu englobe assez clairement une attaque contre $\mathcal{DS}$ et $\mathfrak{H}$ au sens où $\mathcal{A}$ réussit son attaque s'il arrive à retourner soit une collision pour $\mathfrak{H}$ soit une signature d'un message pour lequel aucune requête de signature n'a été faite. Lorsque nous analyserons la sécurité offerte par les protocoles décrits au chapitre 4, nous verrons qu'un attaquant capable de changer la réponse finale sans invalider la ronde d'authentification peut être utilisé pour créer un PPTA gagnant au jeu ci-dessus. Or, l'équation

$$\mathbb{P}\Bigg[\Big((\gamma \neq \sigma) \wedge \big(H_\lambda(\gamma) = H_\lambda(\sigma)\big)\Big) \vee \Big(\Big((\gamma,\sigma) \notin \big\{(m_o,\mathfrak{s}_o)\big\}_o\Big) \wedge \big(Verify_{pk_s}(\gamma,\sigma) = 1\big)\Big)\Bigg] \leq \mathbb{P}\Big[(\gamma \neq \sigma) \wedge \big(H_\lambda(\gamma) = H_\lambda(\sigma)\big)\Big] + \mathbb{P}\Big[\Big((\gamma,\sigma) \notin \big\{(m_o,\mathfrak{s}_o)\big\}_o\Big) \wedge \big(Verify_{pk_s}(\gamma,\sigma) = 1\big)\Big]$$

montre immédiatement que si $\mathcal{DS}$ a des signatures inforgeables et que $\mathfrak{H}$ résiste aux collisions, l'application $Adv^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}}$ est négligeable pour tout PPTA $\mathcal{A}$. Ce sont donc ces hypothèses que nous allons faire sur $\mathcal{DS}$ et $\mathfrak{H}$.

2.4 Récupération confidentielle de données

Ici nous rappelons les travaux de Chor *et al.*, [9]. Il s'agit de récupérer de manière confidentielle une information dans une base de données sans révéler laquelle à la base. Un tel protocole s'appelle un protocole de *récupération confidentielle de données*, abbrévié PIR (*private information retrieval*) dans la suite. Les auteurs montrent comment on peut utiliser la multiplication des bases de données pour pouvoir récupérer une information de manière inconditionnellement confidentielle sans télécharger la base de données toute entière. Dans cette section, $\mathcal{U}$ désigne une entité essayant de récupérér un vecteur binaire $v_i \in \{0,1\}^k$ d'une bases de données contenant N tels vecteurs.

Pour un ensemble E et un élément e, on définit $E \oplus \{e\}$ comme étant $E - \{e\}$ si $e \in E$ et $E \cup \{e\}$ si $e \notin E$.

2.4.1 Deux copies d'une base

Commençons par réviser le protocole de base avec deux bases de données. On suppose qu'il y a deux copies $\mathcal{DB}_0$ et $\mathcal{DB}_1$ d'une même base. Pour récupérer le vecteur v_i, $\mathcal{U}$ commence par choisir uniformément au hasard un sous-ensemble S_0 de $\{1, ..., N\}$ et calcule $S_1 := S_0 \oplus \{i\}$. Il envoie alors S_0 à $\mathcal{DB}_0$ et S_1 à $\mathcal{DB}_1$. La base $\mathcal{DB}_0$ (respectivement, $\mathcal{DB}_1$) calcule alors $X_0 := \bigoplus_{h \in S_0} v_h$ (respectivement, $X_1 := \bigoplus_{h \in S_1} v_h$), et les vecteurs X_0 et X_1 sont renvoyés à $\mathcal{U}$. Enfin, $\mathcal{U}$ calcule $v_i = X_0 \oplus X_1$.

Ce protocole fonctionne car une et une seule des deux parties S_0 et S_1 contient l'indice i, et comme chaque base reçoit une partie uniformément distribuée de $\{1, ..., N\}$, aucune des deux n'a d'information sur i.

2.4.2 Huit copies d'une base

Dans leur travail, Chor *et al.* [9] mettent l'accent sur le fait de diminuer la complexité de la communication du PIR en multipliant la base de données de départ. Nous voyons cette technique plutôt comme un moyen de diminuer l'effort de calcul fait par $\mathcal{U}$. Précisément, nous souhaitons que $\mathcal{U}$ ait à choisir un sous-ensemble de $\{1, ..., N\}$ qui soit plus petit.

On suppose que $N = \ell^3$ pour un certain entier ℓ. On peut toujours se ramener à ce cas par exemple en rajoutant des faux vecteurs à $\mathcal{DB}$. Ainsi, on peut voir la base de données comme un cube de côté ℓ : le vecteur v_i est noté $v_{i^1 i^2 i^3}$ où i^1, i^2 et i^3 sont dans $\{1, ..., \ell\}$. On suppose de plus qu'il y a huit bases de données $\mathcal{DB}_{s_1 s_2 s_3}$ ($s_1 s_2 s_3$ parcourant $\{0,1\}^3$) qui sont chacune une copie de $\mathcal{DB}$, et que $\mathcal{U}$ veut récupérer $v_{i^1 i^2 i^3}$.

Il choisit trois parties S_0^1, S_0^2 et S_0^3 uniformément au hasard dans $\{1, ..., \ell\}$ et calcule les ensembles

$$S_1^1 := S_0^1 \oplus \{i^1\},\ S_1^2 := S_0^2 \oplus \{i^2\} \text{ et } S_1^3 := S_0^3 \oplus \{i^3\}$$

Il envoie ensuite $S_{s_1}^1 \times S_{s_2}^2 \times S_{s_3}^3$ à $\mathcal{DB}_{s_1 s_2 s_3}$ pour tout $s_1 s_2 s_3$, et ce dernier calcule

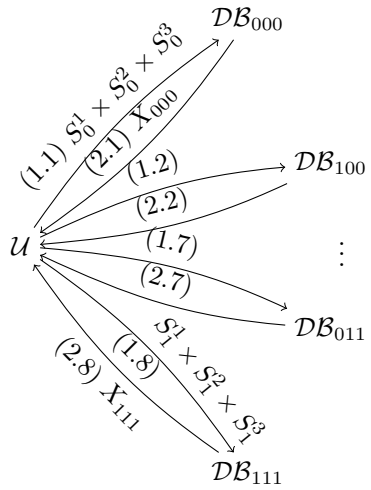
$$X_{s_1 s_2 s_3} := \bigoplus_{S_{s_1}^1 \times S_{s_2}^2 \times S_{s_3}^3} v_{h^1 h^2 h^3}$$

($h^1 h^2 h^3$ parcourant $S_{s_1}^1 \times S_{s_2}^2 \times S_{s_3}^3$) qu'il renvoie à $\mathcal{U}$, qui calcule enfin

$$v_{i^1 i^2 i^3} = \bigoplus_{s_1 s_2 s_3} X_{s_1 s_2 s_3}$$

A nouveau, ce protocole fonctionne simplement parce qu'une et seulement une des parties $S_{s_1}^1 \times S_{s_2}^2 \times S_{s_3}^3$ contient $i^1 i^2 i^3$. Tous les autres vecteurs présents le sont en nombre pair, donc disparaissent. Cependant, $\mathcal{U}$ choisit trois ensembles de taille $\ell = N^{1/3}$ plutôt qu'un ensemble de taille N. Ce PIR est représenté dans la figure 2.2.

FIGURE 2.2 – Un PIR à huit bases de données



2.4.3 Deux copies d'une base en simulant huit

Maintenant, nous allons voir comment deux bases de données peuvent faire le travail de huit d'entre elles afin de diminuer leur nombre. On garde les notations du paragraphe précédent.

L'entité $\mathcal{U}$ choisit encore trois sous-ensembles S_0^1, S_0^2 et S_0^3 dans $\{1, ..., \ell\}$ uniformément au hasard et calcule S_1^1, S_1^2 et S_1^3 comme précédemment. Il envoie alors $S_0^1 \times S_0^2 \times S_0^3$ à $\mathcal{DB}_0$ et $S_1^1 \times S_1^2 \times S_1^3$ à $\mathcal{DB}_1$. La base de données $\mathcal{DB}_0$ calcule X_{000} comme le ferait $\mathcal{DB}_{000}$, et calcule aussi toutes les valeurs possibles pour X_{100}, X_{010} et X_{001}. On comprend mieux ce qui se passe à l'aide d'un exemple. Si on considère X_{100}, $\mathcal{DB}_0$ sait que l'ensemble que $\mathcal{DB}_{100}$ "reçoit" est de la forme $S_0^1 \oplus \{t\} \times S_0^2 \times S_0^3$ pour un certain $t \in \{1, ..., \ell\}$. Par conséquent, $\mathcal{DB}_0$ calcule

$$X_{100}(t) := \bigoplus_{S_0^1 \oplus \{t\} \times S_0^2 \times S_0^3} v_{h^1 h^2 h^3}$$

($h^1 h^2 h^3$ parcourant $S_0^1 \times S_0^2 \times S_0^3$) pour tout tel t. A la fin du calcul, $\mathcal{DB}_0$ a obtenu les $3\ell + 1$ vecteurs binaires

$$X_{100}(1), ..., X_{100}(\ell), X_{010}(1), ..., X_{010}(\ell), X_{001}(1), ..., X_{001}(\ell), X_{000}$$

et de même $\mathcal{DB}_1$ calcule

$$X_{011}(1), ..., X_{011}(\ell), X_{101}(1), ..., X_{101}(\ell), X_{110}(1), ..., X_{110}(\ell), X_{111}$$

Enfin, $\mathcal{DB}_0$ et $\mathcal{DB}_1$ envoient ces vecteurs à $\mathcal{U}$ qui calcule

$$v_{i^1 i^2 i^3} = X_{100}(i^1) \oplus X_{011}(i^1) \oplus X_{010}(i^2) \oplus X_{101}(i^2) \oplus X_{001}(i^3) \oplus X_{110}(i^3) \oplus X_{000} \oplus X_{111}$$

Il est important de remarquer ici que $\mathcal{U}$ utilise le fait qu'il connaît i pour faire le calcul final car il doit choisir les bonnes valeurs à XORer parmi celles qu'il reçoit.

Indiquons rapidement une manière plus efficace que la méthode naïve pour calculer ces XORs. Regardons par exemple $\mathcal{DB}_0$. Ce que l'on désigne par le méthode naïve consisite à calculer, pout tout $t \in \{1, ..., \ell\}$, les trois somme complètes

$$X_{100}(t) := \bigoplus_{S_0^1 \oplus \{t\} \times S_0^2 \times S_0^3} v_{h^1 h^2 h^3},$$

$$X_{010}(t) := \bigoplus_{S_0^1 \times S_0^2 \oplus \{t\} \times S_0^3} v_{h^1 h^2 h^3} \text{ et}$$

$$X_{001}(t) := \bigoplus_{S_0^1 \times S_0^2 \times S_0^3 \oplus \{t\}} v_{h^1 h^2 h^3}$$

en plus de X_{000}. Comme les sous-ensembles S_0^i sont choisis uniformément au hasard, on peut s'attendre à ce qu'ils aient en moyenne $\frac{\ell}{2}$ éléments. Chacune de ces sommes est donc en moyenne le XOR de $\frac{\ell^3}{8}$ vecteurs binaires de longueur k, et il y en a $3\ell + 1$ à calculer ce qui peut rapidement devenir long.

Pour réduire le coût de ce calcul, on observe que comme on travaille modulo 2 on a les formules suivantes pour tout $t \in \{1, ..., \ell\}$:

$$X_{100}(t) = X_{000} \oplus \bigoplus_{S_0^2 \times S_0^3} v_{t h^2 h^3},$$

$$X_{010}(t) = X_{000} \oplus \bigoplus_{S_0^1 \times S_0^3} v_{h^1 t h^3} \text{ et}$$

$$X_{001}(t) = X_{000} \oplus \bigoplus_{S_0^1 \times S_0^2} v_{h^1 h^2 t}$$

Des formules similaires sont vraies pour les calculs que doit faire $\mathcal{DB}_1$ avec X_{111}. Ainsi, plutôt que de calculer $3\ell + 1$ fois des sommes modulo 2 de $\frac{\ell^3}{8}$ vecteurs, on en calcule une, à savoir X_{000} et on calcule les autres à l'aide du vecteur obtenu et des 3ℓ sommes modulo 2 de $\frac{\ell^2}{4}$ (en moyenne) vecteurs données par

$$\bigoplus_{S_0^2 \times S_0^3} v_{t h^2 h^3}, \quad \bigoplus_{S_0^1 \times S_0^3} v_{h^1 t h^3} \text{ et } \bigoplus_{S_0^1 \times S_0^2} v_{h^1 h^2 t}$$

2.4.4 Autres PIR

Il existe bien d'autres protocoles de récupération confidentielle de données, notamment des protocoles à une base de données ayant une sécurité définie en termes de complexité de calcul de manière analogue à ce que nous avons rappelé pour les schémas de chiffrement. Le premier tel protocole a été décrit par

Kushilevitz et Ostrovsky dans [24]. L'un des meilleurs en termes de performances est celui de Lipmaa (voir [27]), d'ailleurs utilisé dans [6] pour améliorer le résultat de [5]. Enfin, citons le protocole de récupération confidentielle de données étendu imaginé par Bringer *et al.* dans [7].

Chapitre 3

Résultats de sécurité théorique

Ce chapitre contient les définitions et résultats relatifs à la sécurité. Il s'agit de rappels la plupart du temps ; toutefois, nous énonçons et démontrons un résultat servant à séparer clairement l'avantage que l'on peut obtenir d'un message chiffré accompagné d'une annexe dépendant du message de celui que l'on obtient uniquement avec l'annexe dans un contexte d'indistinguabilité. Ce résultat est essentiel pour démontrer en partie que notre protocole est sécurisé dans le modèle où l'on se place. Comme certains de nos protocoles opèrent avec plusieurs clefs pour le même schéma de chiffrement, le résultat est établi dans ce cadre.

3.1 Sécurité et indistinguabilité

3.1.1 Introduction

Dans cette section, nous introduisons les notions de sécurité utilisées pour analyser la sécurité des protocoles. Nous rappelons d'abord les notions classiques de sécurité sémantique et d'indistinguabilité des chiffrements. Ces deux notions étant équivalentes et la deuxième étant en général plus commode à manipuler en pratique, nous adoptons les versions "indistinguabilité" des notions de sécurité une fois ce rappel fait.

Nous passons ensuite à l'indistinguabilité sous une attaque à messages clairs choisis ; nous en redonnons la définition et enonçons le résultat permettant de passer à plusieurs clefs. Enfin, en nous plaçant directement dans le cadre de plusieurs clefs, nous introduisons une nouvelle notion de sécurité, l'indistinguabilité à information partielle choisie et information partielle attribuée (notée *ind-cpia-api*-sécurité) et démontrons qu'un cryptosystème est *ind-cpa*-sécurisé si et seulement s'il est sécurisé en ce sens.

La raison d'être de cette caractérisation est de pouvoir séparer clairement les

informations qu'un adversaire peut calculer à partir d'un chiffrement de celles qu'il peut calculer de l'information partielle dans un contexte d'indistinguabilité. Précisément, distinguer entre deux ensembles de messages contenant chacun des chiffrements est ramené à faire une distinction entre ces mêmes messages sans les chiffrements. Cela nous permet notamment de montrer qu'il est impossible en temps raisonnable d'attribuer une identité parmi deux à un ensemble de messages reçus. Le fait que dans notre modèle de sécurité nous autorisons l'adversaire à pouvoir choisir le contenu de la base de données impose que cette caractérisation soit faite en termes de sécurité à messages choisis.

Dans tout ce chapitre, $\mathcal{CS}$ désigne un schéma de chiffrement $(\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ au sens de la définition 7.

3.1.2 Rappels sur la sécurité sémantique

Rappelons la définition de base de la sécurité sémantique. Pour ce faire, nous allons employer le langage des jeux d'attaque. Dans tout ce qui suit, $n \in \mathbb{N}$ est le paramètre de sécurité. Soient $\mathcal{A}$, $\mathcal{A}'$, et $\mathcal{M}$ des PPTA, et $h : \{0,1\}^* \to \{0,1\}^*$ une application polynomialement calculable. Définissons les jeux suivants :

$$
\begin{aligned}
&\mathbf{Exp}^{sem}_{\mathcal{CS},\mathcal{M},h,\mathcal{A}}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (sk, pk) \leftarrow \mathcal{K}(I)\\
&\qquad m \leftarrow \mathcal{M}(I)\\
&\qquad c \leftarrow \mathcal{E}_{pk}(m)\\
&\qquad \gamma \leftarrow \mathcal{A}\big(I, pk, c, 1^{|m|}, h(1^n, m)\big)\\
&\quad \text{Return } \gamma
\end{aligned}
$$

et

$$
\begin{aligned}
&\mathbf{Exp}^{sem}_{\mathcal{P},\mathcal{M},h,\mathcal{A}'}(n)\\
&\qquad m \leftarrow \mathcal{M}(I)\\
&\qquad \gamma' \leftarrow \mathcal{A}'\big(I, 1^{|m|}, h(1^n, m)\big)\\
&\quad \text{Return } \gamma'
\end{aligned}
$$

Soit maintenant $f : \{0,1\}^* \to \{0,1\}^*$ une application polynomialement bornée. On pose :

$$Adv^{f-sem}_{\mathcal{CS},\mathcal{M},h,\mathcal{A}}(n) = \mathbb{P}\big[\gamma = f(1^n, m)\big]$$

et

$$Adv^{f-sem}_{\mathcal{P},\mathcal{M},h,\mathcal{A}'}(n) = \mathbb{P}\big[\gamma' = f(1^n, m)\big]$$

Définition 19 *On dit que $\mathcal{CS}$ est **sémantiquement sécurisé** si pour tout PPTA $\mathcal{A}$ il existe un PPTA $\mathcal{A}'$ tel que pour tout PPTA $\mathcal{M}$ et toute paire d'applications h et f de $\{0,1\}^*$ dans lui-même telles que h soit polynomialement calculable et f soit polynomialement bornée l'application*

$$Adv^{f-sem}_{\mathcal{CS},\mathcal{M},h,\mathcal{A}} - Adv^{f-sem}_{\mathcal{P},\mathcal{M},h,\mathcal{A}'}$$

est négligeable (voir en bas de la page 19 pour la définition de la négligeabilité d'une application).

La définition 19 est la formalisation de la notion intuitive suivante : $\mathcal{CS}$ est sécurisé si l'information que l'on peut calculer en un temps raisonnable avec le chiffrement peut l'être sans le chiffrement. Cette notion est du à Goldwasser et Micali dans [17] et la définition ci-dessus est une formulation en termes de jeux d'attaque de celle que l'on trouve dans [16].

Apparaissant également pour la première fois dans [17] est la notion de cryptosystème ayant des chiffrements indistinguables. Nous en donnons ici une définition tirée encore de [16]. Soit $\mathcal{MP}$ un PPTA retournant des triplets de la forme (m_0, m_1, z) où m_0 et m_1 représentent des messages de même longueur. On définit le jeu :

$$\begin{aligned}
&\mathbf{Exp}^{ind}_{\mathcal{CS},\mathcal{MP},\mathcal{A}}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (sk, pk) \leftarrow \mathcal{K}(I)\\
&\qquad (m_0, m_1, z) \leftarrow \mathcal{MP}(I)\\
&\qquad c_0, c_1 \leftarrow \mathcal{E}_{pk}(m_0), \mathcal{E}_{pk}(m_1)\\
&\qquad \gamma_0, \gamma_1 \leftarrow \mathcal{A}(I, pk, c_0, z), \mathcal{A}(I, pk, c_1, z)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}$$

et on pose

$$Adv^{ind}_{\mathcal{CS},\mathcal{MP},\mathcal{A}}(n) = \big|\mathbb{P}[\gamma_0 = 1] - \mathbb{P}[\gamma_1 = 1]\big|.$$

Avec ces notations, nous avons la

Définition 20 *On dit que $\mathcal{CS}$ a des **chiffrements indistinguables** (ou est ind-sécurisé) si pour tout PPTA $\mathcal{A}$ et tout PPTA $\mathcal{MP}$ (comme ci-dessus) l'application $Adv^{ind}_{\mathcal{CS},\mathcal{MP},\mathcal{A}}$ est négligeable.*

La donnée z qui apparaît dans ce que $\mathcal{MP}$ retourne représente une éventuelle information partielle liée aux messages m_0 et m_1. Il est important de remarquer que la même information est donnée à $\mathcal{A}$ pour distinguer entre des chiffrements de m_0 et m_1.

Un cas particulier important du jeu précédent est celui où l'information z est égale au couple (m_0, m_1). Il va nous être commode dans la suite d'avoir une définition spécifique pour ce cas. Notons $\mathcal{M}$ un PPTA retournant des paires de

messages de même longueur, i.e. de la forme (m_0, m_1) où $|m_0| = |m_1|$. Le jeu devient alors

$$\begin{aligned}
&\mathbf{Exp}^{ind}_{\mathcal{CS},\mathcal{M},\mathcal{A}}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (sk, pk) \leftarrow \mathcal{K}(I)\\
&\qquad (m_0, m_1) \leftarrow \mathcal{M}(I)\\
&\qquad c_0, c_1 \leftarrow \mathcal{E}_{pk}(m_0), \mathcal{E}_{pk}(m_1)\\
&\qquad \gamma_0, \gamma_1 \leftarrow \mathcal{A}(I, pk, c_0, m_0, m_1), \mathcal{A}(I, pk, c_1, m_0, m_1)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}$$

avec comme avantage

$$Adv^{ind}_{\mathcal{CS},\mathcal{M},\mathcal{A}}(n) := \big|\mathbb{P}[\gamma_0 = 1] - \mathbb{P}[\gamma_1 = 1]\big|$$

et cela mène à la

Définition 21 *On dit que $\mathcal{CS}$ a des **chiffrements faiblement indistinguables** (ou est faiblement-ind-sécurisé) si pour tout PPTA $\mathcal{A}$ et tout PPTA $\mathcal{M}$, l'application $Adv^{ind}_{\mathcal{CS},\mathcal{M},\mathcal{A}}$ est négligeable.*

Rappelons maintenant la relation entre ces différentes notions de sécurité. Elle est particulièrement simple :

Proposition 4 *Si $\mathcal{CS}$ est sémantiquement sécurisé, il est ind-sécurisé.*

Voir [16] pour la preuve. □

Ensuite, le lemme suivant est trivial :

Lemme 3 *Si $\mathcal{CS}$ est ind-sécurisé, il est faiblement ind-sécurisé.*

En effet, on a déjà remarqué qu'il suffit de particulariser au cas où $\mathcal{MP}$ retourne des données de la forme $\big((m_0, m_1, (m_0, m_1)\big)$. □

Enfin, on a la proposition frappante suivante, qui permet de boucler la boucle.

Proposition 5 *Si $\mathcal{CS}$ est faiblement-ind-sécurisé, il est sémantiquement sécurisé.*

Voir [16]. □

Ainsi, on a en fait équivalence entre ces trois notions :

Théorème 5 *Le système $\mathcal{CS}$ est ind-sécurisé si et seulement s'il est faiblement-ind-sécurisé.*

3.1.3 Messages clairs choisis et multiples clefs

Une définition à priori plus forte que l'indistinguabilité ci-dessus est l'indistinguabilité à messages clairs choisis. Dans la définition ci-dessus, une paire de messages est donnée à l'attaquant ; pour renforcer l'attaquant et donc avoir une notion de sécurité augmentée, on permet à l'attaquant de choisir lui-même les messages dont il juge pouvoir distinguer les chiffrements. On considère le jeu d'attaque suivant, ou $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ est un PPTA opérant en deux étapes :

$$\begin{aligned}
&\mathbf{Exp}^{ind-cpa}_{\mathcal{CS},\mathcal{A}}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (sk, pk) \leftarrow \mathcal{K}(I)\\
&\qquad (m_0, m_1, e) \leftarrow \mathcal{A}_1(I, pk)\\
&\qquad c_0, c_1 \leftarrow \mathcal{E}_{pk}(m_0), \mathcal{E}_{pk}(m_1)\\
&\qquad \gamma_0, \gamma_1 \leftarrow \mathcal{A}_2(m_0, m_1, e, c_0), \mathcal{A}_2(m_0, m_1, e, c_1)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}$$

et où on pose

$$Adv^{ind-cpa}_{\mathcal{CS},\mathcal{A}}(s) := \big|\mathbb{P}[\gamma_0 = 1] - \mathbb{P}[\gamma_1 = 1]\big|$$

Pour éviter des distinctions trivialement possibles il est supposé ci-dessus que les deux messages retournés par $\mathcal{A}_1$ sont de même longueur. On a alors la

Définition 22 *On dit que $\mathcal{CS}$ est ind-cpa-**sécurisé** si pour tout PPTA $\mathcal{A}$ comme ci-dessus l'application $Adv^{ind-cpa}_{\mathcal{CS},\mathcal{A}}$ est négligeable.*

En pratique, un schéma de chiffrement à clefs publiques comme ci-dessus opère dans un contexte où plusieurs clefs publiques sont publiées. Il est donc important d'analyser dans quelle mesure le nombre de clefs publiques connues affecte la sécurité globale. Cette analyse est faite dans [1], où il est entre autres démontré que dès qu'un chiffrement est *ind-cpa*-sécurisé, il est "*ind-cpa*-sécurisé avec un nombre polynomial de clefs en jeu". Pour avoir une définition précise, il faut définir un jeu d'attaque pour tout polynôme q positif tel que $q(\mathbb{N}) \subset \mathbb{N}$:

$$\begin{aligned}
&\mathbf{Exp}^{ind-cpa-mk}_{q,\mathcal{CS},\mathcal{A}}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (sk_i, pk_i)_i \leftarrow \big(\mathcal{K}(I)\big)_i\\
&\qquad \big((m_{0i})_i, (m_{1i})_i, e\big) \leftarrow \mathcal{A}_1\big(I, (pk_i)_i\big)\\
&\qquad (c_{0i})_i, (c_{1i})_i \leftarrow \big(\mathcal{E}_{pk_i}(m_{0i})\big)_i, \big(\mathcal{E}_{pk_i}(m_{1i})\big)_i\\
&\qquad \gamma_0 \leftarrow \mathcal{A}_2\big((m_{0i})_i, (m_{1i})_i, e, (c_{0i})_i\big)\\
&\qquad \gamma_1 \leftarrow \mathcal{A}_2\big((m_{0i})_i, (m_{1i})_i, e, (c_{1i})_i\big)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}$$

où on pose

$$Adv^{ind-cpa}_{q,\mathcal{CS},\mathcal{A}}(n) := \big|\mathbb{P}[\gamma_0 = 1] - \mathbb{P}[\gamma_1 = 1]\big|$$

et où il est sous-entendu que i parcourt $\{1, ..., q(n)\}$ et que pour tout i, $|m_{0i}| = |m_{1i}|$. La définition cherchée est alors

Définition 23 *On dit que $\mathcal{CS}$ est ind-cpa-mk-**sécurisé** si pour tout PPTA $\mathcal{A}$ et tout polynôme q comme ci-dessus l'application $Adv^{ind-cpa}_{q,\mathcal{CS},\mathcal{A}}$ est négligeable.*

Ce qui est démontré dans [1] peut maintenant être énoncé ainsi :

Théorème 6 *Le schéma de chiffrement $\mathcal{CS}$ est ind-cpa-mk-sécurisé si et seulement s'il est ind-cpa-sécurisé.*

3.2 Informations partielles

Conservons pour cette section les notations de celle d'avant. Si q est un polynôme positif (ne pas oublier la convention), on désignera par $\mathcal{M}_q$ un PPTA servant à générer des messages. Précisément, $\mathcal{M}_q$ prend entrée (I, z_b, z) (où I provient du générateur de paramètres généraux $\mathcal{P}(1^n)$ de $\mathcal{CS}$ et b prendra la valeur 0 ou 1 plus loin) et retourne un $q(n)$-uplet de la forme $(m_{bi})_i$ pour i parcourant $\{1, ..., q(n)\}$. On fait l'hypothèse suivante : pour tout (z_0, z_1) et z

$$|z_0| = |z_1| \Rightarrow \forall\ (m_{0i})_i, (m_{1i})_i \in \mathcal{M}_q(I, z_0, z), \mathcal{M}_q(I, z_1, z),\ \ \forall\ i\ \ |m_{0i}| = |m_{1i}|$$

C'est une hypothèse analogue à celle faite pour la première partie d'un adversaire $\mathcal{A}$ attaquant $\mathcal{CS}$ dans une attaque à messages clairs choisis.

On considèrera aussi un PPTA $\mathcal{PI}$ servant à générer de l'information partielle. Il prend en entrée $(m_{bi})_i$ calculé par $\mathcal{M}_q$ ainsi que l'entrée z et retourne y_b. Il n'est pas nécessaire de faire d'hypothèse supplémentaire sur $\mathcal{PI}$.

Nous allons avoir besoin de deux jeux d'attaque et d'un deuxième PPTA $\mathcal{A}' := (\mathcal{A}'_1, \mathcal{A}'_2)$ opérant en deux temps pour donner une définition précise de la généralisation. Les jeux sont les suivants :

$$\begin{aligned}
&\mathbf{Exp}^{ind-cpia-api-mk}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{P},\mathcal{A}}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (sk_i, pk_i)_i \leftarrow \big(\mathcal{K}(I)\big)_i\\
&\qquad (z_0, z_1, z, e) \leftarrow \mathcal{A}_1\big(I, (pk_i)_i\big)\\
&\qquad (m_{0i})_i, (m_{1i})_i \leftarrow \mathcal{M}_q(I, z_0, z), \mathcal{M}_q(I, z_1, z)\\
&\qquad y_0, y_1 \leftarrow \mathcal{PI}\big((m_{0i})_i, z\big), \mathcal{PI}\big((m_{1i})_i, z\big)\\
&\qquad (c_{0i})_i, (c_{1i})_i \leftarrow \big(\mathcal{E}_{pk_i}(m_{0i})\big)_i, \big(\mathcal{E}_{pk_i}(m_{1i})\big)_i\\
&\qquad \gamma_0 \leftarrow \mathcal{A}_2\big(z_0, z_1, z, e, y_0, (c_{0i})_i\big)\\
&\qquad \gamma_1 \leftarrow \mathcal{A}_2\big(z_0, z_1, z, e, y_1, (c_{1i})_i\big)\\
&\text{Return } (\gamma_0, \gamma_1)
\end{aligned}$$

et

$$\begin{aligned}
&\mathbf{Exp}^{ind-cpia-api-mk}_{q,\mathcal{P},\mathcal{M}_q,\mathcal{PI},\mathcal{A}'}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (z_0, z_1, z, e') \leftarrow \mathcal{A}'_1(I)\\
&\qquad (m_{0i})_i, (m_{1i})_i \leftarrow \mathcal{M}_q(I, z_0, z), \mathcal{M}_q(I, z_1, z)\\
&\qquad y_0, y_1 \leftarrow \mathcal{PI}\big((m_{0i})_i, z\big), \mathcal{PI}\big((m_{1i})_i, z\big)\\
&\qquad \gamma'_0 \leftarrow \mathcal{A}'_2\big(z_0, z_1, z, e', (1^{|m_{0i}|})_i, y_0\big)\\
&\qquad \gamma'_1 \leftarrow \mathcal{A}'_2\big(z_0, z_1, z, e', (1^{|m_{1i}|})_i, y_1\big)\\
&\text{Return } (\gamma'_0, \gamma'_1)
\end{aligned}$$

avec les avantages respectifs

$$Adv^{ind-cpia-api-mk}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{PI},\mathcal{A}}(n) := \big|\mathbb{P}[\gamma_0 = 1] - \mathbb{P}[\gamma_1 = 1]\big|$$

et

$$Adv^{ind-api-cpia-mk}_{q,\mathcal{P},\mathcal{M}_q,\mathcal{PI},\mathcal{A}'}(n) := \big|\mathbb{P}[\gamma'_0 = 1] - \mathbb{P}[\gamma'_1 = 1]\big|$$

où on suppose que i parcourt $\{1, ..., q(n)\}$ et que z_0 et z_1 ont même longueur. La définition qui nous intéresse est alors la

Définition 24 *On dit que $\mathcal{CS}$ a des* ***chiffrements indistinguables sous une attaque à informations partielles choisies, informations partielles attribuées et clefs multiples,*** *ou est ind-cpia-api-mk-sécurisé, si pour tout PPTA $\mathcal{A}$ et tout polynôme q, il existe un PPTA $\mathcal{A}'$ comme ci-dessus tel que pour tout PPTA $\mathcal{M}_q$ et tout PPTA $\mathcal{PI}$ comme ci-dessus, l'application*

$$Adv^{ind-cpia-api-mk}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{PI},\mathcal{A}} - Adv^{ind-api-cpia-mk}_{q,\mathcal{P},\mathcal{M}_q,\mathcal{PI},\mathcal{A}'}$$

est négligeable.

Avant de poursuivre, expliquons le contenu intuitif de cette définition. Il n'importe pas dans ce paragraphe qu'il y ait plusieurs clefs publiques donc on laisse tomber q et mk temporairement.

Le générateur de messages $\mathcal{M}$ prend en entrée des informations choisies par l'adversaire. C'est en cela qu'il s'agit d'une attaque à information partielle choisie (*cpia*). On voit que si $\mathcal{A}_1$ ne retourne rien et que $\mathcal{M}$ est choisi de telle sorte qu'il puisse tourner sans avoir besoin d'entrée provenant de $\mathcal{A}$, on retombe sur une situation d'indistinguabilité où l'adversaire n'a aucun contrôle sur les messages dont il est chargé de faire une distinction des chiffrements. A l'autre extrême, si $\mathcal{M}$ retourne exactement ce qu'il reçoit de $\mathcal{A}_1$, on se trouve dans une situation où $\mathcal{A}_1$ choisit essentiellement les messages à distinguer. Cette définition contient donc tous les intermédiaires entre une indépendance totale des messages et de $\mathcal{A}_1$ (*ind*-sécurité passive) et un contrôle complet de ces messages (*ind-cpa*-sécurité).

Examinons le rôle de $\mathcal{PI}$. L'idée ici est que seule une information relative à un message chiffré et accompagnant le chiffrement de ce message est exploitable pour pouvoir le distinguer du chiffrement d'un autre message accompagné d'une information partielle analogue. Ces informations étant spécifiques aux messages qu'elles accompagnent, on dit qu'elles sont attribuées à ces messages (*api*). Regardons le cas extrême : si $\mathcal{PI}$ retourne une constante quel que soit le message qui est chiffré, $\mathcal{A}_2$ reçoit cette même constante en guise d'information supplémentaire et ne gagne donc rien d'un challenge à un autre. Au niveau des définitions, ce qui disparaît est le terme $Adv^{ind-cpia-api}_{\mathcal{P},\mathcal{M},\mathcal{PI},\mathcal{A}'}$ car y_0 étant égal à y_1 les distributions $\mathcal{A}'_2(z_0, z_1, z, e', y_0)$ et $\mathcal{A}'_2(z_0, z_1, z, e', y_1)$ sont identiques.

En faisant des combinaisons des cas extrêmes décrits ci-dessus, on retrouve des notions de sécurité classiques. Par exemple, si $\mathcal{M}$ retourne exactement ce que $\mathcal{A}_1$ retourne et $\mathcal{PI}$ retourne une constante, on retombe sur la notion d'*ind-cpa*-sécurité (ou d'*ind-cpa-mk*-sécurité).

Le résultat principal de cette section peut maintenant être énoncé.

Proposition 6 *Le cryptosystème $\mathcal{CS}$ est ind-cpa-mk-sécurisé si et seulement s'il est ind-cpia-api-mk-sécurisé.*

Le sens facile de cette proposition, à savoir que l'*ind-cpia-api-mk*-sécurité entraîne l'*ind-cpa-mk*-sécurité, est essentiellement le fait que la première de ces deux notions soit une généralisation de la seconde. Démontrons donc le sens inverse.

Soient q un polynôme et $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ un PPTA jouant contre $\mathcal{CS}$ dans le jeu $\mathbf{Exp}^{ind-cpia-api-mk}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{P},\mathcal{A}}$ pour des PPTA $\mathcal{M}_q$ et $\mathcal{PI}$ comme ci-dessus. On construit $\mathcal{A}' := (\mathcal{A}'_1, \mathcal{A}'_2)$ comme suit.

Pour tout $n \in \mathbb{N}$, $\mathcal{A}'_1$ prend 1^n en entrée, invoque $\mathcal{P}(1^n)$ pour obtenir I, invoque $\big(\mathcal{K}(I)\big)_i$ pour obtenir $(sk_i, pk_i)_i$, invoque $\mathcal{A}_1\big(I, (pk_i)_i\big)$ pour obtenir (z_0, z_1, z, e), pose $e' = e$ et retourne (z_0, z_1, z, e'). Pour $\beta \in \{0, 1\}$, soient maintenant $(m_{\beta i})_i \in \mathcal{M}_q(I, z_\beta, z)$ et $y_\beta \in \mathcal{PI}\big((m_{\beta i})_i, z\big)$; $\mathcal{A}'_2$ prend en entrée $\big(z_0, z_1, z, e', (1^{|m_{\beta i}|})_i, y_\beta\big)$, extrait $(pk_i)_i$ de e', invoque $\big(\mathcal{E}_{pk_i}(1^{|m_{\beta i}|})\big)_i$ pour obtenir $(d_i)_i$, invoque $\mathcal{A}_2\big(z_0, z_1, z, e', y_\beta, (d_i)_i\big)$ pour obtenir γ, pose $\gamma' = \gamma$ et retourne γ'.

On observe que $\mathcal{A}'$ est bien un PPTA car il n'utilise que des PPTA comme sous-programmes. Par ailleurs, $\mathcal{A}'$ ne dépend que de $\mathcal{A}$ et q. Il nous reste donc à montrer que sous l'hypothèse que $\mathcal{CS}$ est *ind-cpa-mk*-sécurisé, l'assertion sur les avantages est vérifiée.

Raisonnons par l'absurde : supposons qu'il existe des PPTA $\mathcal{M}_q$ et $\mathcal{PI}$, un polynôme positif p et un ensemble infini $\mathbb{I}$ d'entiers naturels tels que pour tout $n \in \mathbb{I}$,

$$Adv^{ind-cpia-api-mk}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{PI},\mathcal{A}}(n) - Adv^{ind-cpia-api-mk}_{q,\mathcal{P},\mathcal{M}_q,\mathcal{PI},\mathcal{A}'}(n) > \frac{1}{p(n)}$$

Avec les notations des jeux $\mathbf{Exp}^{ind-cpia-api-mk}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{P},\mathcal{A}}$ et $\mathbf{Exp}^{ind-cpia-api-mk}_{q,\mathcal{P},\mathcal{M}_q,\mathcal{PI},\mathcal{A}'}$ on

réécrit l'inégalité précédente ainsi :

$$\Big|\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_0,(c_{0i})_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(c_{1i})_i\big)=1\Big]\Big|>$$
$$\Big|\mathbb{P}\Big[\mathcal{A}'_2\big(z_0,z_1,z,e',(1^{|m_{0i}|})_i,y_0\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}'_2\big(z_0,z_1,z,e',(1^{|m_{1i}|})_i,y_1\big)=1\Big]\Big|+\frac{1}{p(n)}$$

On peut supposer que pour tout $n\in\mathbb{I}$, on a l'inégalité

$$\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_0,(c_{0i})_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(c_{1i})_i\big)=1\Big]>$$
$$\mathbb{P}\Big[\mathcal{A}'_2\big(z_0,z_1,z,e',(1^{|m_{0i}|})_i,y_0\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}'_2\big(z_0,z_1,z,e',(1^{|m_{1i}|})_i,y_1\big)=1\Big]+\frac{1}{p(n)}$$

Vu la construction de $\mathcal{A}'$ on a pour tout $n\in\mathbb{I}$

$$\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_0,(c_{0i})_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(c_{1i})_i\big)=1\Big]>$$
$$\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e',y_0,(d_i)_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e',y_1,(d_i)_i\big)=1\Big]+\frac{1}{p(n)}$$

i.e. pour tout $n\in\mathbb{I}$

$$\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_0,(c_{0i})_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e',y_0,(d_i)_i\big)=1\Big]>$$
$$\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(c_{1i})_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e',y_1,(d_i)_i\big)=1\Big]+\frac{1}{p(n)}\quad(*)$$

Maintenant, pour exploiter cette inégalité nous allons construire deux autres PPTA $\mathcal{A}^{(j)}:=(\mathcal{A}_1^{(j)},\mathcal{A}_2^{(j)})$ pour $j=0,1$. Pour tout $n\in\mathbb{N}$, tout $I\in\mathcal{P}(1^n)$ et tout $(sk_i,pk_i)_i\in\big(\mathcal{K}(I)\big)_i$, $\mathcal{A}_1^{(j)}$ prend en entrée $\big(I,(pk_i)_i\big)$, invoque $\mathcal{A}_1\big(I,(pk_i)_i\big)$ pour obtenir (z_0,z_1,z,e), invoque $\mathcal{M}_q(I,z_j,z)$ pour obtenir $(m_{ji})_i$, invoque $\mathcal{PI}\big((m_{ji})_i,z\big)$ pour obtenir y_j, pose $e^{(j)}=(z_0,z_1,z,e)$ et retourne

$$\big((m_{ji})_i,((1^{|m_{ji}|})_i,e^{(j)}\big)$$

Puis, pour tout $\big((m_{ji})_i,(1^{|m_{ji}|})_i,e^{(j)}\big)\in\mathcal{A}_1^{(j)}\big(I,(pk_i)_i\big)$, tout

$$(\alpha_i)_i\in\big\{(m_{ji})_i,(1^{|m_{ji}|})_i\big\}$$

et tout $(c_i)_i\in\big(\mathcal{E}_{pk_i}(\alpha_i)\big)_i$, $\mathcal{A}_2^{(j)}$ prend en entrée $\big((m_{ji})_i,((1^{|m_{ji}|})_i,e^{(j)},(c_i)_i\big)$, extrait (z_0,z_1,z,e,y_j) de $e^{(j)}$, invoque $\mathcal{A}_2\big(z_0,z_1,z,y_j,e,(c_i)_i\big)$ pour obtenir γ, et retourne γ.

Il est clair que les $\mathcal{A}^{(j)}$ sont des PPTA jouant contre $\mathcal{CS}$ dans les jeux $\mathbf{Exp}^{ind-cpa-mk}_{q,\mathcal{CS},\mathcal{A}^{(j)}}$. Comme $\mathcal{CS}$ est supposé *ind-cpa-mk*-sécurisé, si n est assez grand on a

$$Adv^{ind-cpa-mk}_{q,\mathcal{CS},\mathcal{A}^{(j)}}(n)\leq\frac{1}{2p(n)}$$

mais par ailleurs on a par construction des $\mathcal{A}^{(j)}$ que

$$\left|\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_j,(c_{ji})_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_j,(d_i)_i\big)=1\Big]\right| = \left|\mathbb{P}\Big[\mathcal{A}_2^{(j)}\big((m_{ji})_i,(1^{|m_{ji}|})_i,e^{(j)},(c_{ji})_i\big)=1\Big]- \mathbb{P}\Big[\mathcal{A}_2^{(j)}\big((m_{ji})_i,(1^{|m_{ji}|})_i,e^{(j)},(d_i)_i\big)=1\Big]\right| = Adv^{ind-cpa-mk}_{q,\mathcal{CS},\mathcal{A}^{(j)}}(n)$$

Ceci nous donne en particulier

$$\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_0,(c_{0i})_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_0,(d_i)_i\big)=1\Big]\leq\frac{1}{2p(n)}$$

et

$$\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(d_i)_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(c_{1i})_i\big)=1\Big]\leq\frac{1}{2p(n)}$$

Enfin, en prenant $n\in\mathbb{I}$ suffisamment grand et en utilisant $(*)$ il vient

$$\frac{1}{2p(n)}>\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(c_{1i})_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(d_i)_i\big)=1\Big]+\frac{1}{p(n)}$$

donc

$$\frac{1}{2p(n)}<\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(d_i)_i\big)=1\Big]-\mathbb{P}\Big[\mathcal{A}_2\big(z_0,z_1,z,e,y_1,(c_{1i})_i\big)=1\Big]$$

d'où la contradiction $\frac{1}{2p(n)}<\frac{1}{2p(n)}$. Le quadruplet $(\mathcal{M}_q,\mathcal{PI},p,\mathbb{I})$ ne peut donc exister et la proposition est démontrée. □

3.2.1 Reformulation des énoncés

Pour appliquer la proposition ci-dessus dans un cadre plus pratique il est commode d'utiliser une formulation différente. On conserve les notations du paragraphe précédent et on considère les jeux d'attaque suivants :

$\mathbf{Exp}^{IND-CPIA-API-MK}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{PI},\mathcal{A}}(n)$

$I\leftarrow\mathcal{P}(1^n)$

$(sk_i,pk_i)_i\leftarrow\big(\mathcal{K}(I)\big)_i$

$(z_0,z_1,z,e)\leftarrow\mathcal{A}_1\big(I,(pk_i)_i\big)$

$\beta\leftarrow\{0,1\}$

$(m_{\beta i})_i\leftarrow\mathcal{M}_q(I,z_\beta,z)$

$y_\beta\leftarrow\mathcal{PI}\big((m_{\beta i})_i,z\big)$

$(c_{\beta i})_i\leftarrow\big(\mathcal{E}_{pk_i}(m_{\beta i})\big)_i$

$\gamma_\beta\leftarrow\mathcal{A}_2\big(z_0,z_1,z,e,y_\beta,(c_{\beta i})_i\big)$

Return γ

et

$$\begin{aligned}
&\mathbf{Exp}^{IND-CPIA-API-MK}_{q,\mathcal{P},\mathcal{M}_q,\mathcal{PI},\mathcal{A}'}(n)\\
&\qquad I \leftarrow \mathcal{P}(1^n)\\
&\qquad (z_0, z_1, z, e') \leftarrow \mathcal{A}'_1(I)\\
&\qquad \beta' \leftarrow \{0,1\}\\
&\qquad (m_{\beta' i})_i \leftarrow \mathcal{M}_q(I, z'_\beta, z)\\
&\qquad y_{\beta'} \leftarrow \mathcal{PI}\big((m_{\beta' i})_i, z\big)\\
&\qquad \gamma' \leftarrow \mathcal{A}'_2\big(z_0, z_1, z, e', (1^{|m_{\beta' i}|})_i, y_{\beta'}\big)\\
&\quad \text{Return } \gamma'
\end{aligned}$$

avec les avantages respectifs

$$Adv^{IND-CPIA-API-MK}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{PI},\mathcal{A}}(n) := \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|$$

et

$$Adv^{IND-CPIA-API-MK}_{q,\mathcal{P},\mathcal{M}_q,\mathcal{PI},\mathcal{A}'}(n) := \big|\mathbb{P}[\gamma' = \beta'] - \frac{1}{2}\big|$$

où i parcourt $\{1, ..., q(n)\}$ et z_0 et z_1 ont même longueur.

Définition 25 *On dit que $\mathcal{CS}$ est IND-$CPIA$-API-MK-**sécurisé** si pour tout PPTA $\mathcal{A}$ et tout polynôme q, il existe un PPTA $\mathcal{A}'$ comme ci-dessus tel que pour tout PPTA $\mathcal{M}_q$ et tout PPTA $\mathcal{PI}$ comme ci-dessus, l'application*

$$Adv^{IND-CPIA-API-MK}_{q,\mathcal{CS},\mathcal{M}_q,\mathcal{PI},\mathcal{A}} - Adv^{IND-CPIA-API-MK}_{q,\mathcal{P},\mathcal{M}_q,\mathcal{PI},\mathcal{A}'}$$

est négligeable.

Bien entendu, on a alors

Proposition 7 *Le cryptosystème $\mathcal{CS}$ est ind-cpa-mk-sécurisé si et seulement s'il est IND-$CPIA$-API-MK-sécurisé.*

□

Remarque : On peut démontrer que l'*ind-cpa*-sécurité est en fait *équivalente* à la sécurité sémantique (voir [16]).

Chapitre 4

Présentation des protocoles

4.1 Introduction

Nous avons maintenant tous les outils nécessaires pour pouvoir présenter et analyser les deux protocoles d'authentification imaginés. Ces protocoles, ainsi que les protocoles intermédiaires auquels nous aurons affaire, sont de la forme

$$\mathcal{PAM} = \left(\mathcal{R}_1, \mathcal{R}_2, \mathcal{AS}, \{\mathcal{C}\}_{\mathcal{C}}, \textsc{Infra}, \textsc{Enr}\right)$$

(voir la définition 2) donc nous fixons une fois pour tout cette notation.

Le premier protocole s'appuye sur le chiffrement ElGamal et peut être théoriquement utilisé avec n'importe quelle famille de groupes pourvu que celle-ci vérifie l'hypothèse DDH, mais nous avons en tête une implémentation avec des groupes de points rationnels de courbes elliptiques définis sur un corps fini. Le chiffrement se fait directement avec les éléments du groupe et cela nous permet de profiter d'une propriété d'homomorphie. Cette dernière fournit un moyen "intrinsèque" pour effectuer un PIR au sein d'une ronde d'authentification.

Le deuxième protocole trouve sa génèse dans le principal défaut du premier, à savoir son coût en terme de calcul au niveau du lecteur. On verra en effet que pour faire une authentification avec le premier il faut que le lecteur fasse appel à un générateur de nombres pseudo-aléatoire autant de fois qu'il y a d'objets enregistrés dans la base. Comme on peut imaginer qu'une telle base de données contiennent jusqu'à 10^9 objets enregistrés, il est clair qu'une méthode plus efficace est souhaitable. Le deuxième protocole tente de remédier à cela en introduisant un protocole PIR séparé du mode de chiffrement, mais à condition d'autoriser plusieurs copies de la base de données. Enfin, ce protocole ne fait appel à aucune propriété algébrique du mode de chiffrement et donc n'importe quel chiffrement par bloc (raisonnablement sécurisé) peut être utilisé.

Malheureusement, ce protocole, noté $\mathcal{PAM}^{final}$ et présenté en 4.3.4, a une sécurité mal comprise au niveau des messages transmis entre $\mathcal{AS}$ et les composantes techniques (dans ce cas, il s'agit de deux copies d'une même base de données). Spécifiquement, nous n'avons pas réussi à démontrer que $\mathcal{PAM}^{final}$

garantit l'anonymat d'objet au niveau de $\mathcal{AS}$ et nous n'avons pu montrer que $\mathcal{PAM}^{final}$ garantit l'anonymat d'objet contre les adversaires extérieurs qu'en rajoutant du chiffrement. Une explication de ce phénomène se trouve en 4.3.4.

Conformément aux définitions que nous avons mises en place dans le chapitre 1, pour décrire les protocoles d'authentification nous devons simplement spécifier quelles sont les composantes $\mathcal{C}$, les fonctionnalités INFRA, ENR et l'interaction entre $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{AS}$ et les $\mathcal{C}$. Ce sera fait au début de chaque paragraphe décrivant un nouveau protocole. La description des $\mathcal{C}$, de INFRA et de ENR est en général assez succinte ; ces données seront fournies ensemble. La procédure d'authentification - étant ce que nous souhaitons vraiment sécuriser et longue à décrire - est mise à part.

Les analyses de sécurité sont toutes de la forme suivante. Un attaquant joue contre le système dans un des jeux d'attaque $\mathbf{Exp}_{\mathcal{PAM},\mathcal{A}}$ définis au chapitre 1 selon les spécifications du protocole d'authentification étudié. L'avantage de l'attaquant est alors mis en relation avec l'avantage d'un attaquant contre un schéma de chiffrement pour les questions de confidentialité, et un schéma de signature et une fonction de hachage pour la question de l'intégrité de la réponse finale. Enfin, la sécurité supposée de ces dernières primitives permet de conclure que l'avantage de l'adversaire de départ contre le système est négligeable.

Lorsqu'il s'agira de prouver des résultats de confidentialité, nous ignorerons complètement la fonction de hachage et le schéma de signature. Il est facile de voir que cela ne nuit pas au résultat. Par contre, pour que des algorithmes puissent faire des requêtes dans les preuves d'intégrité de la réponse finale, il faut que les schémas de chiffrement soient en place bien qu'ils ne servent formellement à rien dans ce cadre sécuritaire.

Nous attirons l'attention du lecteur sur la présence d'un interlude se trouvant dans l'analyse du deuxième protocole. Il s'agit d'un principe utile pour simplifier les démonstrations lors des analyses de sécurité au niveau de la confidentialité. Cela ne concerne pas le premier protocole dont la sécurité ne repose pas sur les résultats du chapitre 3.

Enfin, lorsqu'on manipulera un objet $\mathcal{O}$ à authentifier, il sera souvent commode de noter $f_{i\mathcal{O}}$ plutôt que $f_{\mathcal{O}}$ leur morphométrie listée dans la base de données, $i_{\mathcal{O}}$ étant en général l'indice auquel le flux est stocké.

4.2 Un protocole utilisant ElGamal bit-à-bit

4.2.1 Introduction

Ici nous présentons le premier protocole que nous avons étudié. Il s'agit d'une adaption directe du protocole qui se trouve dans [5] et qui a servi de fil conducteur pour nos recherches. Les composantes du système sont les mêmes ; ce qui change est le mode de chiffrement : à la place du cryptosystème de Goldwasser et Micali (voir [17] ou [5]) nous utilisons le cryptosystème ElGamal bit-à-bit décrit au chapitre 2 dans le paragraphe 2.2.2. Ce choix est fait pour essayer de

diminuer la taille des clefs utilisées. Cependant, l'efficacité du protocole souffre des faits que le chiffrement se fait bit-à-bit et que l'identité de l'objet doit être protégé.

Dans toute cette section, on se donne une famille de groupes

$$\mathfrak{G} := \left\{(q_\lambda, \mathbb{G}_\lambda)\right\}_\lambda,$$

un générateur d'instance $\mathcal{G}$ pour $\mathfrak{G}$ et on note

$$\mathcal{CS} := \mathcal{CS}_{\mathcal{G}}^{EGbb} = (\mathcal{P}, \mathcal{K}, \mathcal{E}^{EGbb}, \mathcal{D}^{EGbb})$$

le cryptosystème ElGamal bit-à-bit associé à $\mathcal{G}$ comme défini en 2.2.2.

On se donne aussi un schéma de signature digitale

$$\mathcal{DS} := (\mathcal{P}_s, \mathcal{K}_s, Sign, Verify)$$

ayant des signatures inforgeables au sens de la définition 16 et une famille de fonctions de hachage

$$\mathfrak{H} := \{H_\mu\}_\mu$$

résistant aux collisions au sens de la définition 18.

4.2.2 Composantes structurelles

Conformément à nos définitions générales, il nous faut spécifier les composantes techniques $\mathcal{C}$ et les deux fonctionnalités Enr et Infra. Soit $n \in \mathbb{N}$ non nul le paramètre de sécurité :

• $\{\mathcal{C}\}_\mathcal{C} = \{\mathcal{DB}, \mathcal{M}\}$: les composantes techniques se trouvant en communication avec le serveur central $\mathcal{AS}$ sont une base de données $\mathcal{DB}$ contenant les flux morphométriques de référence et un comparateur $\mathcal{M}$ utilisé pour prendre la décision finale quand à l'authenticité de l'objet ;

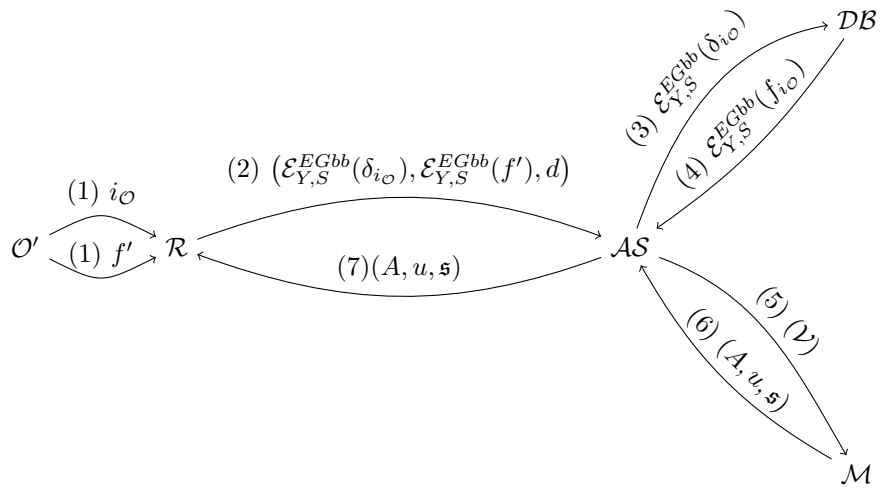
• Infra(1^n) invoque une fois $\mathcal{P}$ pour obtenir un groupe cyclique $\mathbb{G}_\lambda$ d'ordre q_λ ainsi qu'un générateur G de ce groupe. Il invoque ensuite $\mathcal{K}(q_\lambda, \mathbb{G}_\lambda, G)$ pour obtenir la paire de clefs de chiffrement et de déchiffrement $\big((Y,S),x\big)$. Il publie (Y,S), dont $\mathcal{R}$ prend donc connaissance, et donne x à $\mathcal{M}$ qui le garde pour lui-même.

Il invoque ensuite une fois $\mathcal{P}_s(1^n)$ pour obtenir I_s puis $\mathcal{K}_s(I_s)$ pour obtenir les clefs de signature (pk_s, sk_s), donne sk_s à $\mathcal{AS}$ et publie pk_s. En particulier, $\mathcal{R}$ possède une copie de pk_s.

Il invoque $\mathcal{GH}(1^n)$ pour obtenir $\big(\mu, \langle H_\mu\rangle\big)$ et donne ceci à $\mathcal{AS}$ et $\mathcal{R}$.

Enfin, les compteurs $C_{\mathcal{AS}}$ et $C_\mathcal{R}$ de $\mathcal{AS}$ et $\mathcal{R}$ respectivement sont initialisés à 0 ;

• Enr prend en entrée une base de données morphométrique $\big(k, N, \tau, (i, f_i)_i\big)$ et en distribue une copie à $\mathcal{DB}_0$ et $\mathcal{DB}_1$. Il donne de plus (k, N) à $\mathcal{R}$ et τ à $\mathcal{AS}$.



4.2.3 La ronde d'authentification

Pour tester l'authenticité d'un objet $\mathcal{O}'$ s'identifiant comme étant $\mathcal{O}$, et indexé par $i_\mathcal{O}$ dans la base de données, le protocole se déroule comme suit :

1) $\mathcal{R}$ extrait f' et $i_\mathcal{O}$ de $\mathcal{O}'$;

2) $\mathcal{R}$ calcule $\mathcal{E}^{EGbb}_{Y,S}(\delta_{i_\mathcal{O}})$ et $\mathcal{E}^{EGbb}_{Y,S}(f')$ (où $\delta_{i_\mathcal{O}}$ est le vecteur binaire ayant un 1 à la $i_\mathcal{O}$-ième composante et des 0 partout ailleurs). Il incrémente son compteur $C_\mathcal{R}$, et calcule $u := H_\mu\big(\mathcal{E}^{EGbb}_{Y,S}(\delta_{i_\mathcal{O}})||\mathcal{E}^{EGbb}_{Y,S}(f')||C_\mathcal{R}\big)$. Enfin, il envoie

$$\left(\mathcal{E}^{EGbb}_{Y,S}(\delta_{i_\mathcal{O}}), \mathcal{E}^{EGbb}_{Y,S}(f')\right)$$

à $\mathcal{AS}$ tout en gardant u en mémoire ;

3) $\mathcal{AS}$ envoie $\mathcal{E}^{EGbb}_{Y,S}(\delta_{i_\mathcal{O}})$ à $\mathcal{DB}$;

4) $\mathcal{DB}$ calcule pour tout entier h entre 0 et $k-1$, l'élément

$$\sum_{v=1}^{N} f_{vh}\mathcal{E}^{EGbb}_{Y,S}(\delta_{i_\mathcal{O}v})$$

Grâce à l'homomorphie cet élément n'est autre que $\mathcal{E}^{EGbb}_{Y,S}(f_{i_\mathcal{O}h})$ (on donne les détails de ce calcul plus loin) donc à la fin de cette étape $\mathcal{DB}$ a en fait calculé $\mathcal{E}^{EGbb}_{Y,S}(f_{i_\mathcal{O}})$. Il envoie cet élément à $\mathcal{AS}$;

5) $\mathcal{AS}$ calcule $\mathcal{E}^{EGbb}_{Y,S}(f_{i_\mathcal{O}}) + \mathcal{E}^{EGbb}_{Y,S}(f')$ qui - toujours par homomorphie - n'est autre que $\mathcal{E}^{EGbb}_{Y,S}(f' \oplus f_{i_\mathcal{O}})$. Il choisit alors uniformément au hasard une permutation σ de $\{0, ..., k-1\}$ qu'il applique chiffrement par chiffrement au vecteur

d'éléments du groupe $\mathcal{E}^{EGbb}_{Y,S}(f' \oplus f_{i_\mathcal{O}})$. Notant le vecteur obtenu $\mathcal{V}$, $\mathcal{AS}$ incrémente son compteur $C_{\mathcal{AS}}$, calcule $u := H_\mu\big(\mathcal{E}^{EGbb}_{Y,S}(\delta_{i_\mathcal{O}})||\mathcal{E}^{EGbb}_{Y,S}(f')||C_{\mathcal{AS}}\big)$ et envoie $\mathcal{V}$ à $\mathcal{M}$;

6) $\mathcal{M}$ déchiffre $\mathcal{V}$. Il calcule alors le poids de Hamming du vecteur qu'il obtient (qui est clairement celui du vecteur avant application de la permutation), le compare à τ, détermine sa réponse A et l'envoie à $\mathcal{AS}$;

7) Le serveur $\mathcal{AS}$ invoque $Sign_{sk_s}(A, u)$ pour obtenir une signature digitale $\mathfrak{s}$ de (A, u) et transmet $(A, u, \mathfrak{s})$ à $\mathcal{R}$;

8) Le lecteur $\mathcal{R}$ vérifie la validité de la signature digitale, et vérifie que le u qu'il reçoit est le même que celui qu'il a calculé avant d'envoyer ses messages à $\mathcal{AS}$. Si l'une de ces vérifications échoue, il rejette la réponse.

4.2.4 Coût en calculs

Evaluons l'effort fournit par les diverses composantes. Rappelons que k désigne la longueur des flux morphométriques et N le nombre d'objets enregistrés dans la base de données.

• Le lecteur $\mathcal{R}$ doit chiffrer bit-à-bit un vecteur binaire de longueur $k + N$. Vu les spécifications du protocole de chiffrement, il doit faire appel à un générateur pseudo-aléatoire d'entiers de l'ordre de la centaine de bits $k + N$ fois. Soit $\mathfrak{l}$ la longueur des entiers choisis. Le lecteur doit aussi calculer le hachage d'un vecteur de longueur au plus $\mathfrak{l}(k+N)$. Il doit maintenir en mémoire un compteur $C_\mathcal{R}$. Enfin, il doit vérifier une signature digitale. L'effort du lecteur est linéaire en la taille de la base de données et des flux morphométriques.

Au total, pour générer le premier message, le lecteur doit choisir $\mathfrak{l}(k + N)$ bits uniformément au hasard, et envoyer au serveur un message de longueur $\mathfrak{l}(k + N)$.

• Le serveur d'authentification doit calculer le hachage d'un vecteur de taille au plus $\mathfrak{l}(k+N)$. Il doit calculer $2k$ sommes de deux éléments du groupe $\mathbb{G}_\lambda$. Il doit générer de manière pseudo-aléatoire une permutation de $\{0, ..., k - 1\}$. Il doit maintenir en mémoire un compteur $C_{\mathcal{AS}}$. Enfin, il doit calculer une signature digitale.

• La base de données doit calculer la somme d'au plus $2N$ éléments du groupe $\mathbb{G}_\lambda$.

• Le comparateur $\mathcal{M}$ doit calculer k déchiffrements ElGamal.

4.2.5 Intégrité de la réponse finale contre les adversaires extérieurs

Commençons par établir que l'intégrité de la réponse finale est garantie contre des adversaires extérieurs observant le canal de communication entre

$\mathcal{R}$ et $\mathcal{AS}$. Pour ce faire, il faut réduire l'attaque contre le schéma d'authentification à une attaque contre soit le schéma de signature, soit la fonction de hachage.

Théorème 7 *Si le schéma de signature $\mathcal{DS}$ a des signatures inforgeables et la famille de fonctions de hachage $\mathfrak{H}$ résiste aux collisions, le schéma d'authentification $\mathcal{PAM}$ garantit l'intégrité de la réponse d'une ronde d'authentification contre les adversaires extérieurs.*

Soit $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4)$ un PPTA opérant en quatre étapes et jouant contre $\mathcal{PAM}$ dans le jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{B}}(n)$ pour n un paramètre de sécurité. On va construire un PPTA $\mathcal{A}$ opérant en deux temps et utilisant $\mathcal{B}$ comme sous-programme pour soit trouver une collision pour $\mathfrak{H}$, soit forger une signature pour $\mathcal{DS}$.

- *Phase de mise en place par $\mathcal{CH}$ des paramètres pour $\mathcal{A}$*

Le challengeur $\mathcal{CH}$ invoque $\mathcal{P}_s(1^n)$ pour obtenir I_s puis $\mathcal{K}_s(I_s)$ pour obtenir les clefs (pk_s, sk_s) du schéma de signature. Ensuite, il invoque $\mathcal{GH}(1^n)$ pour obtenir $\big(\mu, \langle H_\mu \rangle\big)$ de la famille $\mathfrak{H}$. Il envoie $\big(pk_s, \langle H_\mu \rangle\big)$ à $\mathcal{A}_1$.

- *Phase de mise en place par $\mathcal{A}$ des paramètres pour $\mathcal{B}$*

$\mathcal{A}_1$ invoque $\mathcal{P}(1^n)$ pour obtenir $(\lambda, \mathbb{G}_\lambda, G)$ et invoque ensuite $\mathcal{K}$ pour obtenir les clefs du chiffrement ElGamal bit-à-bit $\big((Y,S),x\big)$.

- *Phase d'appels à l'oracle de signature par $\mathcal{A}$*

Les requêtes de signature que $\mathcal{A}_1$ fait dans cette phase sont utilisées pour répondre aux requêtes de rondes d'authentification que fait $\mathcal{B}_1$. Le PPTA $\mathcal{A}$ invoque $\mathcal{B}_1\big(pk_s, (Y,S), \langle H_\mu \rangle\big)$. Il répond aux requêtes d'authentification de $\mathcal{B}_1$ comme suit.

Lorsque $\mathcal{B}_1$ spécifie une o-ième requête de la forme

$$\Big(j_o, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big),$$

$\mathcal{A}_1$ simule tous les calculs visibles par l'adversaire du protocole d'une ronde d'authentification normale avec base de données $\big(k_o, N_o, \tau_o, (i, f_{oi})_i\big)$, flux de référence f_{oj_o} et flux extrait $f_{oj_o} \oplus \kappa_o$. Autrement dit, $\mathcal{A}_1$ commence par invoquer $\mathcal{E}^{EGbb}_{Y,S}(\delta_{j_o})$ et $\mathcal{E}^{EGbb}_{Y,S}(f_{oj_o} \oplus \kappa_o)$ pour obtenir $c_{o\mathcal{DB}} \in \mathbb{G}_\lambda^{2N_o}$ et $c_{o\mathcal{AS}} \in \mathbb{G}_\lambda^{2k_o}$ respectivement. Ensuite, il incrémente le compteur du lecteur qui doit donc valoir o. Puis, il détermine A_o en fonction de κ_o et τ_o, incrémente le compteur $C_{\mathcal{AS}}$ du serveur d'authentification qui vaut donc o également, pose $c_o := c_{o\mathcal{AS}} || c_{o\mathcal{DB}}$, calcule $u_o := H_\mu(c_o||o)$, et envoie (A_o, u_o) comme requête de signature à $\mathcal{CH}$. Ce dernier invoque alors $Sign_{sk_s}(A_o, u_o)$ pour obtenir $\mathfrak{s}_o$ qui est retourné à $\mathcal{A}_1$. Enfin, $\mathcal{A}_1$ répond à $\mathcal{B}_1$ en lui donnant $\Big(o, c_o, \big((A_o, u_o), \mathfrak{s}_o\big)\Big)$.

A la fin de la phase d'appels à l'oracle de ronde, $\mathcal{A}_1$ et $\mathcal{B}_1$ ont la liste suivante :

$$\mathfrak{Q} := \left(\Big(j_o, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o \Big), \Big(o, c_o, \big(\big(j(\kappa_o, \tau_o), u_o\big), \mathfrak{s}_o \big) \Big) \right)_o$$

pour o parcourant $\{1, ..., q\}$ où q est majoré par un polynôme en n. La première partie d'un tel vecteur est l'information de la requête et la deuxième le résultat. Ce dernier contient comme réponse finale $A_o = j(\kappa_o, \tau_o)$ car les rondes sont valides.

Il n'est pas supposé que les requêtes soient deux-à-deux distinctes mais l'attaquant n'aura jamais de réponses identiques à cause du compteur o.

Pour conclure cette phase, $\mathcal{B}_1$ retourne la liste $\mathfrak{Q}$ ci-dessus et une information d'état e_1.

- *Requête de challenge de $\mathcal{B}$*

  $\mathcal{A}_1$ invoque $\mathcal{B}_2(\mathfrak{Q}, e_1)$ pour obtenir une requête de challenge

$$\Big(j, \big(k, N, \tau, (i, f_i)_i\big), \kappa, e_2 \Big)$$

Cette requête peut déjà apparaître dans $\mathfrak{Q}$.

- *Première phase de réponse au challenge : $\mathcal{A}$ simule le premier message de $\mathcal{R}$*

  $\mathcal{A}_1$ invoque $\mathcal{E}^{EGbb}_{Y,S}(\delta_j)$ et $\mathcal{E}^{EGbb}_{Y,S}(f_j \oplus \kappa)$ pour obtenir respectivement $c_{\mathcal{DB}}$ et $c_{\mathcal{AS}}$, incrémente $C_{\mathcal{R}}$ qui vaut donc $q+1$, pose $c = c_{\mathcal{AS}} || c_{\mathcal{DB}}$ et calcule $u := H_\mu\big(c||(q+1)\big)$.

- *Deuxième phase de réponse au challenge : $\mathcal{B}$ manipule les messages de $\mathcal{R}$ vers $\mathcal{AS}$*

  $\mathcal{A}_1$ invoque $\mathcal{B}_3(c_{\mathcal{AS}}, c_{\mathcal{DB}}, e_2)$ pour obtenir $(c'_{\mathcal{AS}}, c'_0, e_3)$ de sorte que $c'_{\mathcal{AS}}$ et $c'_{\mathcal{DB}}$ soient de la forme

$$(c'_{\mathcal{AS}h})_{h\in\{0,...,k-1\}} \in \mathbb{G}_\lambda^{2k} \text{ et } (c'_{\mathcal{DB}h})_{h\in\{1,...N\}} \in \mathbb{G}_\lambda^{2N}$$

respectivement et que pour tout $h \in \{0, ..., k-1\}$

$$c'_{\mathcal{AS}h} + \sum_{v=1}^{N} f_{vh} \mathcal{E}^{EGbb}_{Y,S}(\delta_{jv}) \in \big\{ \mathcal{E}^{EG}_Y(0), \mathcal{E}^{EG}_Y(S), \mathcal{E}^{EG}_Y(2S) \big\}$$

On met cette hypothèse sur la forme des messages chiffrés que l'attaquant peut injecter afin de ne pas invalider la ronde pour des raisons triviales (e.g., si $\mathcal{M}$ obtient comme déchiffrement mS pour $m \geq 3$ ou si les messages ont la mauvaise longueur).

- *Troisième phase de réponse au challenge : $\mathcal{A}$ simule les calculs de l'interaction entre $\mathcal{AS}$ et les composantes $\mathcal{DB}$ et $\mathcal{M}$*

$\mathcal{A}_1$ calcule

$$c'_h := c'_{\mathcal{AS}h} + \sum_{v=1}^{N} f_{vh}\mathcal{E}^{EGbb}_{Y,S}(\delta_{jv}) \in \{\mathcal{E}^{EG}_Y(0), \mathcal{E}^{EG}_Y(S), \mathcal{E}^{EG}_Y(2S)\}$$

pour tout h. Ensuite $\mathcal{A}_1$ choisit une permutation σ de $\{0, ..., k-1\}$ uniformément au hasard, calcule

$$\mathcal{V} := (c'_{\sigma(h)})_h,$$

déchiffre $\mathcal{V}$ pour obtenir κ'_σ, calcule le poids de Hamming de κ'_σ, calcule A' en fonction de ce poids, pose $c' = c'_{\mathcal{AS}}||c'_{\mathcal{DB}}$ et calcule $u' := H_\mu\big(c'||(q+1)\big)$ (car le compteur $C_{\mathcal{AS}}$ affiche $(q+1)$). Puis, il envoie (A', u') à $\mathcal{CH}$.

• *Quatrième phase de réponse au challenge : $\mathcal{A}$ obtient de $\mathcal{CH}$ une signature digitale pour simuler le message de $\mathcal{AS}$ à $\mathcal{R}$*

$\mathcal{CH}$ invoque $Sign_{sk_s}(A', u')$ pour obtenir $\mathfrak{s}'$ et envoie $\mathfrak{s}'$ à $\mathcal{A}_2$.

• *Réponse au challenge de $\mathcal{B}$: $\mathcal{B}$ manipule les messages de $\mathcal{AS}$ vers $\mathcal{R}$*

$\mathcal{A}_2$ invoque $\mathcal{B}_4\big((A', u'), \mathfrak{s}', e_3\big)$ pour obtenir $\big((A'', u''), \mathfrak{s}''\big)$ où $A'' \in \{0, 1\}$.

• *Réponse au challenge de $\mathcal{A}$: $\mathcal{A}$ cherche dans ce qu'il a calculé une collision ou une signature forgée*

Si $A'' = j(\kappa, \tau) \oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'', u''), \mathfrak{s}''\big) = 1$, $\mathcal{A}_2$ pose $cm_{\mathcal{R}_2} = A''$. Puis :

- si $(A'', u'') = (A', u')$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c'||(q+1)\big)$ et
- si $(A'', u'') \neq (A', u')$:
  - si $(A'', u'') \notin \big\{(A_o, u_o)\big\}_o$, $\mathcal{A}$ retourne (A'', u'') et
  - s'il existe o tel que $(A'', u'') = (A_o, u_o)$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c_o||o\big)$.

Dans tous les autres cas, $\mathcal{A}_2$ retourne un message d'échec.

Ceci achève la description de $\mathcal{A}$. Il faut maintenant établir formellement la relation entre les avantages. D'une part, les trois conditions $A'' = j(\kappa, \tau) \oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'', u''), s''\big) = 1$ sont satisfaites si et seulement si $cm_{\mathcal{R}_2} = j(\kappa, \tau) \oplus 1$ à la fin du jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{B}}(n)$. D'autre part, si ces conditions sont satisfaites on a soit $c' \neq c$ soit $(A'', u'') \neq (A', u')$ car dans le cas contraire, l'adversaire $\mathcal{B}$ est un PPTA ne modifiant aucun message du challenge et la réponse finale de l'authentification doit être $j(\kappa, \tau)$. Il en résulte que

• si $(A'', u'') = (A', u')$, $c' \neq c$ et $\big(c'||(q+1), c||(q+1)\big)$ est une collision pour H_μ car $u'' = u$ et $u'' = u'$ donc

$$H_\mu\big(c'||(q+1)\big) = u' = u = H_\mu\big(c||(q+1)\big)$$

et

• si $(A'', u'') \neq (A', u')$, l'équation $Verify_{pk_s}\big((A'', u''), \mathfrak{s}''\big) = 1$ signifie que $\mathfrak{s}''$ est une signature valide sur le message (A'', u'') et il faut maintenant voir si (A'', u'') a déjà été demandée par $\mathcal{A}$ à $\mathcal{CH}$ ou non.

Si $(A'', u'') \neq (A_o, u_o)$ pour tout o, (A'', u'') ne fait pas partie des signatures demandées à $\mathcal{CH}$ et donc c'est une vraie signature valide forgée par $\mathcal{A}$.

S'il existe o tel que $(A'', u'') = (A_o, u_o)$, on a nécessairement $c||(q+1) \neq c_o||o$ car $o \leq q$. On obtient donc à nouveau une collision car

$$H_\mu(c_o||o) = u_o = u'' = u = H_\mu\big(c||(q+1)\big)$$

Vu que $\mathcal{A}$ joue contre $\mathfrak{H}$ et $\mathcal{DS}$ dans le jeu $\mathbf{Exp}^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}}$, on conclut :

$$Adv^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}} \geq Adv^{int-ext}_{\mathcal{PAM},\mathcal{B}}$$

et la négligeabilité du membre de gauche entraîne celle du membre de droite. □

Remarque : Les preuves des énoncés concernant l'intégrité pour les autres protocoles sont quasiment identiques à celle ci-dessus. Les différences essentielles sont dans les phases de requêtes et de challenge lors de l'interaction entre $\mathcal{B}$ et $\mathcal{A}$ car celles-ci doivent être conformes aux spécifications des protocoles.

4.2.6 Confidentialité

Pour considérablement simplifier les preuves des résultats concernant l'anonymat de l'objet lors d'un ronde d'authentification on remarque la chose suivante. Le serveur d'authentification et la base de données n'ont pas la clef de déchiffrement selon les spécifications du paramètrage et les manipulations qu'ils font sont faisables par n'importe quel adversaire extérieur dans le modèle où nous nous sommes placés. On en déduit que pour démontrer les assertions concernant la confidentialité d'une authentification au niveau d'un adversaire extérieur, du serveur $\mathcal{AS}$ et de la base de données $\mathcal{DB}$, **il suffit de le faire pour l'adversaire extérieur.** En effet, un tel adversaire reçoit comme information les messages entre toutes les composantes. Or, les composantes ne peuvent utiliser comme informations que celles qu'elles reçoivent lors d'une ronde valide. Ces messages formant un sous-ensemble des messages reçus par l'adversaire extérieur et $\mathcal{AS}$ et $\mathcal{DB}$ ne faisant pas de calculs qui restent secrets pour eux, ils n'ont pas plus de pouvoir que les adversaires extérieurs.

Par contre, il faut raisonner différemment pour le comparateur $\mathcal{M}$ car il possède la clef secrète de déchiffrement, ce qui n'est bien entendu pas le cas d'un adversaire extérieur, et non plus le cas de $\mathcal{AS}$ et $\mathcal{DB}$.

Enfin on rappelle que les portions du protocole employant la fonction de hachage et la signature digitale sont omises. Cela ne nuit pas au résultat car ces primitives ne concernent pas la confidentialité.

Adversaire extérieurs, serveur d'authentification malicieux et base de données malicieuse

Notre résultat principal dans ce paragraphe est le suivant.

Théorème 8 *Si le générateur $\mathcal{G}$ de la famille de groupes $\mathfrak{G}$ satisfait l'hypothèse DDH, le schéma d'authentification $\mathcal{PAM}$ garantit l'anonymat d'objet contre les adversaires extérieurs, un serveur d'authentification malicieux et une base de données malicieuse.*

C'est une conséquence de la proposition suivante et des résultats du paragraphe 2.2.2.

Proposition 8 *Si $\mathcal{CS}_{\mathcal{G}}^{EGbb}$ est ind-cpa-sécurisé, le schéma d'authentification $\mathcal{PAM}$ garantit l'anonymat d'objet contre les adversaires extérieurs, un serveur d'authentification malicieux et une base de données malicieuse.*

Comme nous l'avons expliqué plus haut, les messages que reçoivent $\mathcal{AS}$ et $\mathcal{DB}$ lors d'une ronde d'authentification forment un sous-ensemble de ceux reçus par l'adversaire extérieur, et ils n'ont pas plus de pouvoir de calcul que ce dernier d'après les paramètres. Il nous suffit donc de raisonner pour l'adversaire extérieur.

Soient n un paramètre de sécurité et $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$ un PPTA jouant contre le schéma d'authentification $\mathcal{PAM}$ dans le jeu $\mathbf{Exp}^{oa-ext}_{\mathcal{PAM},\mathcal{B}}(n)$. Nous allons construire un PPTA $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ interagissant avec un challengeur $\mathcal{CH}$ pour attaquer $\mathcal{EG}_{\mathcal{G}}^{EGbb}$ en utilisant $\mathcal{B}$ comme sous-programme.

- *Mise en place des paramètres par $\mathcal{CH}$ pour $\mathcal{A}$ et par $\mathcal{A}$ pour $\mathcal{B}$*

Le challengeur $\mathcal{CH}$ invoque $\mathcal{P}(1^n)$ pour obtenir $(q_\lambda, \mathbb{G}_\lambda, G)$ puis $\mathcal{K}(q_\lambda, \mathbb{G}_\lambda, G)$ pour obtenir une paire de clefs $\big((Y,S),x\big)$ pour le schéma de chiffrement $\mathcal{CS}_{\mathcal{G}}^{EGbb}$. Il envoie $\big((Y,S),x\big)$ à $\mathcal{A}_1$ qui invoquera $\mathcal{B}_1$ avec.

- *Phase de création d'une demande de challenge pour $\mathcal{A}$ par $\mathcal{B}$*

L'attaquant $\mathcal{A}_1$ invoque $\mathcal{B}_1(Y,S)$ pour obtenir une base de données et deux indices de cette base

$$\Big(i_0, i_1, \big((k, N, \tau, (i, f_i)_i,), \kappa\big), (Y,S)\Big)$$

Ainsi, k et N sont polynomiaux en n car $\mathcal{B}_1$ opère en temps polynomial. Le sous-programme $\mathcal{B}_1$ retourne aussi une information d'état $e_{\mathcal{B}}$.

- *Phase de création d'une demande de challenge pour $\mathcal{CH}$ par $\mathcal{A}$*

L'attaquant $\mathcal{A}_1$ va maintenant utiliser

$$\Big(i_0, i_1, \big((k, N, \tau, (i, f_i)_i,), \kappa\big)\Big)$$

pour créer une paire de messages à envoyer à $\mathcal{CH}$. Spécifiquement, $\mathcal{A}_1$ pose

$$m_0 = f_{i_0} \oplus \kappa || \delta_{i_0} \text{ et } m_1 = f_{i_1} \oplus \kappa || \delta_{i_1}$$

Ces deux vecteurs binaires sont des éléments de $\{0,1\}^{k+N}$. Il pose enfin $e_{\mathcal{A}} = e_{\mathcal{B}}$ et envoie maintenant (m_0, m_1) à $\mathcal{CH}$ et $\big((m_0, m_1), e_{\mathcal{A}}\big)$ à $\mathcal{A}_2$.

- *Phase de création du challenge par $\mathcal{CH}$ pour $\mathcal{A}$*

Le challengeur $\mathcal{CH}$ choisit β uniformément au hasard dans $\{0,1\}$. Ensuite, il invoque $\mathcal{E}^{EGbb}_{Y,S}(m_\beta) \in \mathbb{G}_\lambda^{2(k+N)}$ pour obtenir c et envoie c à $\mathcal{A}_2$ en guise de réponse à la demande de challenge.

- *Phase de création du challenge pour $\mathcal{B}$ par $\mathcal{A}$*

La deuxième partie de l'attaquant, $\mathcal{A}_2$, partitionne c en $c = c_{\mathcal{AS}} || c_{\mathcal{DB}}$, où $c_{\mathcal{AS}} \in \mathbb{G}_\lambda^{2k}$ et $c_{\mathcal{DB}} \in \mathbb{G}_\lambda^{2N}$. Il calcule ensuite pour tout $h \in \{0, ..., k-1\}$

$$c'_{\mathcal{AS}h} + \sum_{v=1}^{N} f_{vh} c_{\mathcal{DB}v}$$

et on voit tout de suite que $c'_{\mathcal{AS}} := (c'_{\mathcal{AS}h})_h \in \mathcal{E}^{EGbb}_{Y,S}(f_{i_\beta})$. Enfin, $\mathcal{A}_2$ pose $d = c_{\mathcal{AS}} + c'_{\mathcal{AS}}$, choisit une permutation σ de $\{0,1\}^k$ uniformément au hasard et pose $\mathcal{V} = \sigma(d)$. La réponse à la demande de challenge de $\mathcal{B}_1$ va être alors

$$(c_{\mathcal{AS}}, c_{\mathcal{DB}}, \mathcal{V})$$

où $c_{\mathcal{AS}} \in \mathcal{E}^{EGbb}_{Y,S}(f_{i_\beta} \oplus \kappa)$, $c_{\mathcal{DB}} \in \mathcal{E}^{EGbb}_{Y,S}(\delta_{i_\beta}$ et $\mathcal{V} \in \mathcal{E}^{EGbb}_{Y,S}\big(\sigma(\kappa)\big)$. On remarque que le dernier de ces vecteurs est obtenu par homomorphie.

- *Phase de réponse de $\mathcal{B}$ pour $\mathcal{A}$ et de $\mathcal{A}$ pour $\mathcal{CH}$*

L'attaquant $\mathcal{A}_2$ invoque $\mathcal{B}_2(c_{\mathcal{AS}}, c_{\mathcal{DB}}, \mathcal{V}, e_{\mathcal{B}})$ pour obtenir une réponse γ. Il envoie alors γ comme réponse à $\mathcal{CH}$ pour la valeur de β.

La construction de $\mathcal{A}$ utilisant $\mathcal{B}$ est achévée. Il faut maintenant regarder ce qui se passe au niveau des avantages des attaquants. Il est clair que $\mathcal{A}$ gagne au jeu $\mathbf{Exp}^{ind-cpa}_{\mathcal{CS}^{EGbb}_{\mathcal{G}},\mathcal{A}}$ si et seulement si $\mathcal{B}$ gagne au jeu $\mathbf{Exp}^{oa-ext}_{\mathcal{PAM},\mathcal{B}}$. On a donc

$$Adv^{ind-cpa}_{\mathcal{CS}^{EGbb}_{\mathcal{G}}} = Adv^{oa-ext}_{\mathcal{PAM},\mathcal{B}}$$

et la négligeabilité du membre de gauche entraîne celle du membre de droite. De même, on montre que les applications $Adv^{oa-comp}_{\mathcal{PAM},\mathcal{AS}}$ et $Adv^{oa-comp}_{\mathcal{PAM},\mathcal{DB}}$ sont négligeables, d'où le résultat. □

Un comparateur malicieux

Il reste à traiter le cas du comparateur. Comme il dispose de la clef secrète, la résultat ci-dessous dépend du fait qu'il n'a accès à aucune autre information lors d'une ronde d'authentification.

Lemme 4 *L'application*

$$Adv^{oa-comp}_{\mathcal{PAM},\mathcal{M}}$$

est identiquement nulle, donc négligeable.

Si $\mathcal{M}_1$ retourne comme base de données

$$\left(i_0, i_1, \left(\left(k, N, \tau, (i, f_i)_i, \right), \kappa\right)\right)$$

en guise de demande de challenge, l'information que $\mathcal{M}_2$ reçoit est (après avoir effectué un déchiffrement) $\sigma(\kappa)$ où σ est une permutation de $\{0, ..., k-1\}$ choisie uniformément au hasard (et d'ailleurs inconnue de $\mathcal{M}$). Comme κ est indépendant du choix de β fait par le challengeur, il est clair que $\sigma(\kappa)$ ne donne aucune information sur β, d'où l'énoncé. □

4.3 Des protocoles à multiples bases de données

4.3.1 Introduction

Dans cette partie, nous décrivons des protocoles d'authentification d'objets à distance construits à partir d'un schéma de chiffrement générique $\mathcal{CS} := (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ sur lequel on ne fait pas d'hypothèse autre que celle concernant sa sécurité. Spécifiquement, $\mathcal{CS}$ est supposé *ind-cpa*-sécurisé, et on a vu au chapitre 3 que cela équivaut à être $IND\text{-}CPIA\text{-}API\text{-}MK$-sécurisé. Comme dans la section précédente, on dispose aussi d'un schéma de signature digitale $\mathcal{DS}$ ayant des signatures inforgeables au sens de la définition 16 ainsi que d'une famille de fonctions de hachage $\mathfrak{H} := \{H_\mu\}_\mu$ résistant aux collisions au sens de la définition 18.

Le protocole principal est le dernier décrit mais pour faciliter sa description et sa preuve de sécurité nous allons l'introduire en le présentant comme une modification de protocoles plus simples pour lesquels des preuves de sécurité sont aussi présentées. Les preuves d'intégrité ont la même forme que celle donnée pour le premier protocole : on suppose l'existence d'un adversaire $\mathcal{B}$ cherchant à modifier les messages de manière à falsifier la réponse et on en déduit un adversaire capable d'attaquer directement la fonction de hachage ou le schéma de signature. Par contre, les preuves de confidentialité sont légèrement différentes. Elles s'appuient sur les résultats qui occupent le chapitre 3 et il nous a semblé plus commode de voir les adversaires contre le protocole directement comme des adversaires contre les schémas de chiffrement afin de mieux appliquer ces résultats. On ne construit donc pas formellement d'adversaire contre les schémas de chiffrement à partir de ceux contre le protocole. On traite les adversaires du protocole plutôt directement comme des adversaires du schéma de chiffrement.

4.3.2 Un protocole simple avec deux bases de données

Ce protocole est celui qui a servi de prototype pour les suivants, plus compliqués. Il contient déjà toutes les idées générales et fait pleinement utilisation des notions de sécurité introduites au chapitre 3.

Composantes structurelles

Définissons les $\mathcal{C}$, INFRA et ENR :

• $\{\mathcal{C}\}_{\mathcal{C}} = \{\mathcal{DB}_0, \mathcal{DB}_1\}$, c'est-à-dire que les composantes techniques sont réduites à deux PPTA ;

• INFRA(1^n) invoque une fois $\mathcal{P}(1^n)$, trois fois indépendantes $\mathcal{K}(I)$ pour obtenir $(pk_{\mathcal{AS}}, sk_{\mathcal{AS}})$, (pk_0, sk_0) et (pk_1, sk_1), distribue $sk_{\mathcal{AS}}$, sk_0 et sk_1 à $\mathcal{AS}$, $\mathcal{DB}_0$ et $\mathcal{DB}_1$ respectivement, et publie $(pk_{\mathcal{AS}}, pk_0, pk_1)$. En particulier, $\mathcal{R}$ reçoit une copie de $(pk_{\mathcal{AS}}, pk_0, pk_1)$.

Il invoque ensuite une fois $\mathcal{P}_s(1^n)$ pour obtenir I_s puis $\mathcal{K}_s(I_s)$ pour obtenir les clefs de signature (pk_s, sk_s), donne sk_s à $\mathcal{AS}$ et publie pk_s. En particulier, $\mathcal{R}$ possède une copie de pk_s.

Il invoque $\mathcal{GH}(1^n)$ pour obtenir $\big(\mu, \langle H_\mu \rangle\big)$ et donne ceci à $\mathcal{AS}$ et $\mathcal{R}$.

Enfin, les compteurs $C_{\mathcal{AS}}$ et $C_{\mathcal{R}}$ de $\mathcal{AS}$ et $\mathcal{R}$ respectivement sont initialisés à 0 ;

• ENR prend en entrée une base de données morphométrique $\big(k, N, \tau, (i, f_i)_i\big)$ et en distribue une copie à $\mathcal{DB}_0$ et $\mathcal{DB}_1$. Il donne de plus (k, N) à $\mathcal{R}$ et τ à $\mathcal{AS}$.

La ronde d'authentification

Décrivons maintenant l'interaction entre les composantes lors d'une ronde d'authentification en fonction des spécifications ci-dessus. Soit $i_{\mathcal{O}} \in \{1, ..., N\}$ et $\mathcal{O}'$ un objet prétendant être $\mathcal{O}$. La précédure se déroule comme suit pour authentifier l'objet :

1) $\mathcal{R}$ lit $i_{\mathcal{O}}$ et extrait f' ;

2) $\mathcal{R}$ choisit deux masques M_0 et M_1 dans $\{0,1\}^k$. Il choisit aussi $S_0 \subseteq \{1, ..., N\}$ et calcule S_1 à l'aide de $i_{\mathcal{O}}$ et S_0. Il calcule ensuite $c_0 := \mathcal{E}_{pk_0}(S_0||M_0)$, $c_1 := \mathcal{E}_{pk_1}(S_1||M_1)$ et $c_{\mathcal{AS}} := \mathcal{E}_{pk_{\mathcal{AS}}}(f' \oplus M_0 \oplus M_1)$. Enfin, il incrémente son compteur $C_{\mathcal{R}}$, calcule $u := H_\mu(c_{\mathcal{AS}}||c_0||c_1||C_{\mathcal{R}})$ qu'il garde en mémoire et envoie $(c_0, c_1, c_{\mathcal{AS}})$ à $\mathcal{AS}$.

3) $\mathcal{AS}$ envoie c_0 et c_1 à $\mathcal{DB}_0$ et $\mathcal{DB}_1$ respectivement.

4) $\mathcal{DB}_0$ (resp., $\mathcal{DB}_1$) calcule S_0 et M_0 (resp., S_1 et M_1). Puis, il calcule :

$$X_0 := M_0 \oplus \bigoplus_{h \in S_0} f_h$$

De même, $\mathcal{DB}_1$ calcule

$$X_1 := M_1 \oplus \bigoplus_{h \in S_1} f_h$$

Ensuite, $\mathcal{DB}_0$ (resp., $\mathcal{DB}_1$) envoie X_0 (resp., X_1) à $\mathcal{AS}$.

5) $\mathcal{AS}$ calcule $f' \oplus M_0 \oplus M_1$. Puis il calcule la somme $f' \oplus M_0 \oplus M_1 \oplus X_0 \oplus X_1$ pour obtenir $f' \oplus f_{i_\mathcal{O}}$. Il détermine le poids w de Hamming du vecteur obtenu et retourne la réponse A selon que $w \leq \tau$ ou $w > \tau$. Il incrémente son compteur $C_{\mathcal{AS}}$, calcule $u := H_\mu(c_{\mathcal{AS}}||c_0||c_1||C_{\mathcal{AS}})$, invoque $Sign_{sk_s}(A, u)$ pour obtenir une signature digitale $\mathfrak{s}$ sur (A, u) et envoie $\big((A, u), \mathfrak{s}\big)$ à $\mathcal{R}$.

6) $\mathcal{R}$ regarde si le u qu'il a gardé en mémoire est égal à celui envoyé par $\mathcal{AS}$. Il vérifie ensuite la signature digitale $\mathfrak{s}$ sur (A, u). Si l'une de ces vérifications échoue, $\mathcal{R}$ retourne $\perp$ et invalide la ronde. Sinon, il accepte A.

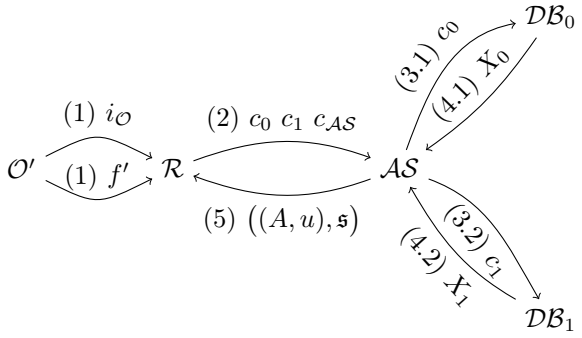


FIGURE 4.1 – Un protocole simple

Coût en calculs

Comme pour le premier protocole, on fait un bilan de l'effort fournit par chaque composante lors d'une ronde d'authentification.

• Le lecteur $\mathcal{R}$ doit générer de manière pseudo-aléatoire deux vecteurs binaires de longueur k. Il doit générer de manière pseudo-aléatoire également un vecteur binaire de longueur N. Il doit calculer le chiffrement d'un message de longueur k et deux chiffrements de messages de longueur N. Si $lg(c_k)$ et $lg(c_{k+N})$ sont les longueurs respectives d'un chiffré d'un message de longueur k et de longueur $k + N$, il doit calculer le hachage d'un message de longueur $lg(c_k) + 2lg(c_{k+N})$. Enfin, il doit vérifier une signature digitale.

Au total, le lecteur doit générer $2k + N$ bits uniformément au hasard en plus de l'aléa nécéssaire au chiffrement de deux messages de longueur k et d'un message de longueur N. Il envoie au serveur d'authentification un message de longueur $lg(c_k) + 2lg(c_{k+N})$.

• Le serveur d'authentification $\mathcal{AS}$ doit déchiffrer deux messages de longueur $lg(c_k)$ et générer une signature digitale. Il doit aussi calculer le hachage d'un message de longueur $lg(c_k)$ et maintenir un compteur $C_{\mathcal{AS}}$.

• Chaque base de données doit déchiffrer un message de longueur $lg(c_{k+N})$ et XORer au plus N flux morphométriques de longueur k.

Adversaires extérieurs

Traitons d'abord le cas des adversaires extérieurs. On rappelle que dans le modèle de sécurité où l'on se place, il s'agit d'un adversaire actif cherchant à changer la réponse finale en observant le canal de communication entre $\mathcal{R}$ et $\mathcal{AS}$ et d'un adversaire passif cherchant à déterminer l'identité à priori de l'objet en observant toutes les communications entre les composantes.

Intégrité

Théorème 9 *Si le schéma de signature $\mathcal{DS}$ a des signatures inforgeables et la famille de fonctions de hachage $\mathfrak{H}$ résiste aux collisions, le schéma d'authentification $\mathcal{PAM}$ garantit l'intégrité de la réponse d'une ronde d'authentification contre les adversaires extérieurs.*

Considérons $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4)$ un PPTA en quatre étapes jouant contre $\mathcal{PAM}$ selon $\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{B}}(n)$ où n est le paramètre de sécurité et construisons un PPTA $\mathcal{A}$ en deux étapes, utilisant $\mathcal{B}$ comme sous-programme et cherchant soit à trouver une collision pour $\mathfrak{H}$, soit à forger une signature pour $\mathcal{DS}$. Comme nous l'avons dit dans la remarque à la fin du paragraphe 4.2.5, la démonstration est presque identique à celle de 7, à la forme des requêtes près.

• *Phase de mise en place par $\mathcal{CH}$ des paramètres pour $\mathcal{A}$*

Le challengeur $\mathcal{CH}$ invoque $\mathcal{P}_s(1^n)$ pour avoir I_s puis $\mathcal{K}_s(I_s)$ pour avoir les clefs (pk_s, sk_s) de $\mathcal{DS}$. Ensuite, il invoque $\mathcal{GH}(1^n)$ pour obtenir un couple $\big(\mu, \langle H_\mu \rangle\big)$ provenant de $\mathfrak{H}$. Il envoie $\big(pk_s, \langle H_\mu \rangle\big)$ à $\mathcal{A}_1$.

• *Phase de mise en place par $\mathcal{A}$ des paramètres pour $\mathcal{B}$*

$\mathcal{A}_1$ invoque $\mathcal{P}(1^n)$ pour obtenir I et invoque ensuite trois fois $\mathcal{K}(I)$ pour créer les clefs de chiffrement $\big((pk_{\mathcal{AS}}, sk_{\mathcal{AS}}), (pk_0, sk_0), (pk_1, sk_1)\big)$.

• *Phase d'appels à l'oracle de signature par $\mathcal{A}$*

$\mathcal{A}_1$ fait des requêtes de signature auprès de $\mathcal{CH}$ pour répondre aux requêtes de rondes que fait $\mathcal{B}_1$. Le PPTA $\mathcal{A}$ invoque $\mathcal{B}_1\big(pk_s, pk_{\mathcal{AS}}, pk_0, pk_1, \langle H_\lambda \rangle\big)$ et répond aux requêtes d'authentification de $\mathcal{B}_1$ comme spécifié ci-dessous.

Lorsque $\mathcal{B}_1$ spécifie une o-ième requête de la forme

$$\Big(j_o, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big),$$

$\mathcal{A}_1$ simule les calculs d'une ronde valide avec base de données de départ

$$\big(k_o, N_o, \tau_o, (i, f_{oi})_i\big),$$

flux de référence f_{oj_o} et flux extrait $f_{oj_o} \oplus \kappa_o$. Autrement dit, $\mathcal{A}_1$ choisit M_{o0} et M_{o1} dans $\{0,1\}^{k_o}$ uniformément au hasard, choisit S_{o0} dans l'ensemble des parties de $\{1, ..., N_o\}$ uniformément au hasard, calcule $S_{o1} := S_{o0} \oplus \{j_o\}$, calcule $g_o := f_{oj_o} \oplus \kappa_o \oplus M_{o0} \oplus M_{o1}$, invoque $\mathcal{E}_{pk_{\mathcal{AS}}}(g_o)$ pour obtenir $c_{o\mathcal{AS}}$, invoque $\mathcal{E}_{pk_0}(M_{o0}||S_{o0})$ pour obtenir c_{o0}, invoque $\mathcal{E}_{pk_1}(M_{o1}||S_{o1})$ pour obtenir c_{o1}, incrémente le compteur du lecteur $C_{\mathcal{R}}$ qui doit donc valoir o, pose $c_o = c_{o\mathcal{AS}}||c_{o0}||c_{o1}$ et calcule $u_o := H_\mu(c_o||o)$. Ensuite, il détermine A_o en fonction de κ_o et τ_o, incrémente le compteur du serveur d'authentification $C_{\mathcal{AS}}$ (qui donc doit valoir o aussi) et envoie (A_o, u_o) comme requête de signature à $\mathcal{CH}$. Le challengeur $\mathcal{CH}$ invoque alors $Sign_{sk_s}(A_o, u_o)$ pour obtenir $\mathfrak{s}_o$ qui est retourné à $\mathcal{A}_1$. Ce dernier répond maintenant à $\mathcal{B}_1$ en lui donnant $\Big(o, c_o, \big((A_o, u_o), \mathfrak{s}_o\big)\Big)$.

Quand cette phase d'appels à l'oracle de ronde se termine, $\mathcal{A}_1$ et $\mathcal{B}_1$ ont la liste de données suivante :

$$\mathfrak{Q} := \left(\Big(j_o, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big), \Big(o, c_o, \Big(\big(j(\kappa_o, \tau_o), u_o\big), \mathfrak{s}_o\Big)\Big)\right)_o$$

pour o parcourant $\{1, ..., q\}$ où q est majoré par un polynôme en n. La première partie moitié de ces données est l'information de la requête et la deuxième le résultat obtenu. La réponse finale est $A_o = j(\kappa_o, \tau_o)$ car les rondes sont valides.

A la fin de la phase $\mathcal{B}_1$ retourne la liste $\mathfrak{Q}$ ci-dessus et une information d'état e_1.

- *Requête de challenge de $\mathcal{B}$*

  $\mathcal{A}_1$ invoque $\mathcal{B}_2(\mathfrak{Q}, e_1)$ pour obtenir cette fois une requête de challenge

$$\Big(j, \big(k, N, \tau, (i, f_i)_i\big), \kappa, e_2\Big)$$

qui peut éventuellement déjà apparaître dans $\mathfrak{Q}$.

- *Première phase de réponse au challenge : $\mathcal{A}$ simule le premier message de $\mathcal{R}$*

  $\mathcal{A}_1$ choisit M_0 et M_1 uniformément au hasard dans $\{0,1\}^k$, choisit S_0 uniformément au hasard parmi les sous-ensembles de $\{1, ..., N\}$, calcule $S_1 := S_0 \oplus \{j\}$, calcule $g := f_j \oplus \kappa \oplus M_0 \oplus M_1$, invoque $\mathcal{E}_{pk_{\mathcal{AS}}}(g)$, $\mathcal{E}_{pk_0}(M_0||S_0)$ et $\mathcal{E}_{pk_1}(M_1||S_1)$ pour obtenir respectivement $c_{\mathcal{AS}}$, c_0 et c_1, incrémente $C_{\mathcal{R}}$ qui vaut donc $q+1$, pose $c = c_{\mathcal{AS}}||c_0||c_1$ et calcule $u := H_\mu\big(c||(q+1)\big)$.

- *Deuxième phase de réponse au challenge : $\mathcal{B}$ manipule les messages de $\mathcal{R}$ vers $\mathcal{AS}$*

  $\mathcal{A}_1$ invoque $\mathcal{B}_3(c_{\mathcal{AS}}, c_0, c_1, e_2)$ pour obtenir $(c'_{\mathcal{AS}}, c'_0, c'_1, e_3)$ de sorte qu'il existe $g' \in \{0,1\}^k$ et L_0 et L_1 dans $\{0,1\}^{k+N}$ vérifiant $c'_{\mathcal{AS}} \in \mathcal{E}_{pk_{\mathcal{AS}}}(g')$,

$c'_0 \in \mathcal{E}_{pk_0}(L_0)$ et $c'_1 \in \mathcal{E}_{pk_1}(L_1)$. Comme dans l'analyse du premier protocole, cette hypothèse sur la forme des messages chiffrés que l'attaquant peut injecter est mise afin de ne pas invalider la ronde pour des raisons triviales : il faut au moins que la longueur des flux soit correcte et que le déchiffrement des messages corrompus fournissent des données permettant aux calculs de l'étape suivante de faire sens.

• *Troisième phase de réponse au challenge : $\mathcal{A}$ simule les calculs de l'interaction entre $\mathcal{AS}$ et les composantes $\mathcal{DB}_0$ et $\mathcal{DB}_1$*

$\mathcal{A}_1$ calcule g', L_0 et L_1 avec les clefs secrètes et partitionne L_0 et L_1 en $M'_0||S'_0$ et $M'_1||S'_1$ respectivement, où M'_0 et M'_1 sont dans $\{0,1\}^k$ et S'_0 et S'_1 sont dans $\{0,1\}^N$. Ensuite, $\mathcal{A}_1$ calcule $X'_0 := M'_0 \oplus \bigoplus_{h\in S'_0} f_h$ et $X'_1 := M'_1 \oplus \bigoplus_{h\in S'_1} f_h$, calcule $\kappa' := g' \oplus X'_0 \oplus X'_1$, calcule le poids de Hamming de κ', calcule A' en fonction de ce poids, pose $c' = c'_{\mathcal{AS}}||c'_0||c'_1$ et calcule $u' := H_\mu\big(c'||(q+1)\big)$ (car le compteur $C_{\mathcal{AS}}$ affiche $(q+1)$). Puis, il envoie (A', u') à $\mathcal{CH}$.

• *Quatrième phase de réponse au challenge : $\mathcal{A}$ obtient de $\mathcal{CH}$ une signature digitale pour simuler le message de $\mathcal{AS}$ à $\mathcal{R}$*

$\mathcal{CH}$ invoque $Sign_{sk_s}(A', u')$ pour calculer $\mathfrak{s}'$ et envoie $\mathfrak{s}'$ à $\mathcal{A}_2$.

• *Réponse au challenge de $\mathcal{B}$: $\mathcal{B}$ manipule les messages de $\mathcal{AS}$ vers $\mathcal{R}$*

$\mathcal{A}_2$ invoque $\mathcal{B}_4\big((A', u'), \mathfrak{s}', e_3\big)$ pour avoir $\big((A'', u''), \mathfrak{s}''\big)$ où $A'' \in \{0,1\}$.

• *Réponse au challenge de $\mathcal{A}$: $\mathcal{A}$ cherche dans ce qu'il a calculé une collision ou une signature forgée*

Si $A'' = j(\kappa,\tau)\oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'', u''), \mathfrak{s}''\big) = 1$, $\mathcal{A}_2$ pose $cm_{\mathcal{R}_2} = A''$. Puis :

- si $(A'', u'') = (A', u')$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c'||(q+1)\big)$ et
- si $(A'', u'') \neq (A', u')$:
  - si $(A'', u'') \notin \big\{(A_o, u_o)\big\}_o$, $\mathcal{A}$ retourne (A'', u'') et
  - s'il existe o tel que $(A'', u'') = (A_o, u_o)$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c_o||o\big)$.

Dans tous les autres cas, $\mathcal{A}_2$ retourne un message d'échec.

La description de $\mathcal{A}$ étant terminée, étudions la relation entre les avantages. On a d'une part que les trois conditions $A'' = j(\kappa,\tau) \oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'', u''), s''\big) = 1$ sont satisfaites si et seulement si $cm_{\mathcal{R}_2} = j(\kappa,\tau)\oplus 1$ à la fin du jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{B}}(n)$, et on a d'autre part que si ces conditions sont satisfaites, soit $c' \neq c$ soit $(A'', u'') \neq (A', u')$ (sinon $\mathcal{B}$ ne modifie aucun message du challenge et la réponse finale de l'authentification doit être $j(\kappa,\tau)$). Ainsi :

• si $(A'', u'') = (A', u')$, $c' \neq c$ et $\big(c'||(q+1), c||(q+1)\big)$ est une collision pour H_μ car $u'' = u$ et $u'' = u'$ donc

$$H_\mu\big(c'||(q+1)\big) = u' = u = H_\mu\big(c||(q+1)\big)$$

et

• si $(A'', u'') \neq (A', u')$, l'équation $Verify_{pk_s}\big((A'', u''), \mathfrak{s}''\big) = 1$ montre que $\mathfrak{s}''$ est une signature valide sur le message (A'', u'') et il reste à voir si (A'', u'') a déjà été demandée par $\mathcal{A}$ lors d'une requête ou non.

Si $(A'', u'') \neq (A_o, u_o)$ pour tout o, (A'', u'') n'est pas dans l'ensemble des signatures demandées à $\mathcal{CH}$ et donc il s'agit d'une signature valide forgée réellement par $\mathcal{A}$.

S'il y a un indice o tel que $(A'', u'') = (A_o, u_o)$, on a nécessairement $c||((q+1) \neq c_o||o$ vu que $o \leq q$. Cela donne donc à nouveau une collision car

$$H_\mu(c_o||o) = u_o = u'' = u = H_\mu\big(c||(q+1)\big)$$

Enfin, $\mathcal{A}$ joue contre $\mathfrak{H}$ et $\mathcal{DS}$ dans le jeu $\mathbf{Exp}^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}}$ donc on conclut :

$$Adv^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}} \geq Adv^{int-ext}_{\mathcal{PAM},\mathcal{B}}$$

Le member de droite est alors négligeable car c'est le cas du membre de gauche. □

Confidentialité

Occupons-nous à présent de la confidentialité.

Théorème 10 *Si le schéma de chiffrement $\mathcal{CS}$ est ind-cpa-sécurisé, le schéma d'authentification $\mathcal{PAM}$ garantit l'anonymat d'objet contre les adversaires extérieurs.*

Soit n un paramètre de sécurité. Pour établir le théorème il faut simuler une interaction entre un challengeur $\mathcal{CH}$ et un PPTA $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ dans le jeu $\mathbf{Exp}^{oa-ext}_{\mathcal{PAM},\mathcal{A}}(n)$ et évaluer l'avantage de l'adversaire. Vu les spécifications du schéma $\mathcal{PAM}$, cette interaction peut être décrite comme suit.

Le challengeur $\mathcal{CH}$ invoque $\mathcal{P}(1^n)$ une fois pour obtenir I puis $\mathcal{K}(I)$ trois fois pour obtenir les clefs publiques $pk_{\mathcal{AS}}$, pk_0 et pk_1. Il les donne à $\mathcal{A}_1$ qui calcule $\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big)\Big)$ et le donne à $\mathcal{CH}$. Il donne également

$$\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big), (pk_{\mathcal{AS}}, pk_0, pk_1)\Big)$$

à $\mathcal{A}_2$. Ensuite, $\mathcal{CH}$ choisit :

- $\beta \in \{0, 1\}$ uniformément au hasard,
- M_0 et M_1 dans $\{0, 1\}^k$ uniformément au hasard et
- $S_0 \subseteq \{1, ..., N\}$ uniformément au hasard,

calcule :

- $S_1 := S_0 \oplus \{i_\beta\}$,
- $X_0 := M_0 \oplus \bigoplus_{h \in S_0} f_h$ et $X_1 := M_1 \bigoplus_{h \in S_1} f_h$ et

- $g := f_{i_\beta} \oplus \kappa \oplus M_0 \oplus M_1$,

invoque :

- $\mathcal{E}_{pk_{\mathcal{AS}}}(g)$ pour obtenir $c_{\mathcal{AS}}$,
- $\mathcal{E}_{pk_0}(M_0||S_0)$ pour obtenir c_0 et
- $\mathcal{E}_{pk_1}(M_1||S_1)$ pour obtenir c_1 et

donne $\big((X_0, X_1), c_{\mathcal{AS}}, c_0, c_1\big)$ à $\mathcal{A}_2$. Vu ce qui lui a été donné par $\mathcal{A}_1$, $\mathcal{A}_2$ a à sa disposition

$$\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big), (pk_{\mathcal{AS}}, pk_0, pk_1), (X_0, X_1), c_{\mathcal{AS}}, c_0, c_1\Big)$$

pour essayer de deviner β. Notons sa réponse γ : on a par définition

$$Adv^{oa-ext}_{\mathcal{PAM},\mathcal{A}}(s) = \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|$$

Maintenant, il est clair que l'interaction ci-dessus entre $\mathcal{CH}$ et $\mathcal{A}$ dans le jeu $\mathbf{Exp}^{oa-ext}_{\mathcal{PAM},\mathcal{A}}$ est en fait une instance du jeu d'attaque $\mathbf{Exp}^{IND-CPIA-API-MK}_{3,\mathcal{CS},\mathcal{M}_3,\mathcal{PI},\mathcal{A}}(n)$ où

- $(z_0, z_1, z) = \Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big)\Big)$,
- $e = (pk_{\mathcal{AS}}, pk_0, pk_1)$,
- $(m_{\beta 1}, m_{\beta 2}, m_{\beta 3}) = (g, M_0||S_0, M_1||S_1)$ et
- $y_\beta = (X_0, X_1)$.

On a alors aussi par définition de $\mathbf{Exp}^{IND-CPIA-API-MK}_{3,\mathcal{CS},\mathcal{M}_3,\mathcal{PI},\mathcal{A}}$ que

$$Adv^{IND-CPIA-API-MK}_{3,\mathcal{CS},\mathcal{M}_3,\mathcal{PI},\mathcal{A}}(s) = \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|$$

donc on a

$$Adv^{oa-ext}_{\mathcal{PAM},\mathcal{A}} = Adv^{IND-CPIA-API-MK}_{3,\mathcal{CS},\mathcal{M}_3,\mathcal{PI},\mathcal{A}}$$

et il nous faut démontrer que le membre de droite dans l'égalité ci-dessus est négligeable en n.

Comme $\mathcal{A}$ est un PPTA et que $\mathcal{CS}$ est *ind-cpa*-sécurisé, il existe un PPTA $\mathcal{A}' := (\mathcal{A}'_1, \mathcal{A}'_2)$ qui opère comme décrit plus bas en interaction avec le challenger $\mathcal{CH}$, et tel que

$$Adv^{IND-CPIA-API-MK}_{3,\mathcal{CS},\mathcal{M}_3,\mathcal{PI},\mathcal{A}} - Adv^{IND-CPIA-API-MK}_{3,\mathcal{P},\mathcal{M}_3,\mathcal{PI},\mathcal{A}'}$$

soit négligeable.

Pour $I \in \mathcal{P}(1^n)$ le PPTA $\mathcal{A}'_1$ prend en entrée I et calcule $\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big)\Big)$ qu'il donne à $\mathcal{CH}$ et à $\mathcal{A}'_2$. Ensuite, $\mathcal{CH}$ choisit :

- $\beta' \in \{0, 1\}$ uniformément au hasard,
- M_0 et M_1 dans $\{0, 1\}^k$ uniformément au hasard et
- $S_0 \subseteq \{1, ..., N\}$ uniformément au hasard,

calcule :

- $S_1 := S_0 \oplus \{i_\beta\}$ et
- $X_0 := M_0 \oplus \bigoplus_{h \in S_0} f_h$ et $X_1 := M_1 \oplus \bigoplus_{h \in S_1} f_h$ et

donne (X_0, X_1) à $\mathcal{A}'_2$. Comme $\mathcal{A}'_1$ a déjà donné $\Big(i_0, i_1, \big(, \kappa, \tau, (i, f_i)_i\big)\Big)$ à $\mathcal{A}'_2$ pour qu'il essaye de deviner β, $\mathcal{A}'_2$ a les informations suivantes :

$$\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big), (X_0, X_1)\Big)$$

Notons sa réponse γ'.

Enfin, vu que M_0 et M_1 sont choisis chacun uniformément au hasard, X_0 et X_1 sont uniformément distribués dans $\{0,1\}^k$. On obtient donc que $Adv^{IND-CPIA-API-MK}_{3,\mathcal{P},\mathcal{M}_3,\mathcal{PI},\mathcal{A}'}$ est identiquement nul, ce qui implique que $Adv^{IND-CPIA-API-MK}_{3,\mathcal{CS},\mathcal{M}_3,\mathcal{PI},\mathcal{A}}$ est négligeable. Le théorème est donc démontré. □

Interlude : principe de simplification des démonstrations

La démonstration donnée ici de ce théorème est très formelle. L'application de la proposition 6 permet d'introduire formellement un algorithme $\mathcal{A}'$ qui opère essentiellement aussi bien que $\mathcal{A}$ mais avec les chiffrements de messages en moins. L'argument final consiste alors à montrer que les informations partielles attribuées n'apportent aucune indication non plus. **Ce qu'il faut retenir du contenu de la proposition 6 est en fin de compte que les chiffrements peuvent être ignorés. Pour allèger les preuves, nous n'introduirons plus les algorithmes $\mathcal{A}'$; nous nous contenterons d'indiquer que seules les informations partielles importent.**

Composantes malicieuses

Théorème 11 *Si le schéma de chiffrement $\mathcal{CS}$ est ind-cpa-sécurisé, le schéma d'authentification $\mathcal{PAM}$ garantit l'anonymat d'objet contre les composantes serveur malicieuses.*

Par construction du schéma $\mathcal{PAM}$, le théorème est conséquence des deux lemmes suivants.

Lemme 5 *Si $\mathcal{CS}$ est ind-cpa-sécurisé et $\mathcal{CM} = \mathcal{AS}$, l'application*

$$Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$$

est négligeable.

Comme pour le théorème 10, il faut utiliser les spécifications concrètes de $\mathcal{PAM}$ pour simuler une instance du jeu $\mathbf{Exp}^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$ et évaluer l'avantage de $\mathcal{CM} = \mathcal{AS}$. Le jeu se décrit comme suit.

Soit n un paramètre de sécurité. Le challengeur $\mathcal{CH}$ invoque $\mathcal{P}(1^n)$ pour obtenir I et $\mathcal{K}(I)$ deux fois pour obtenir les clefs publiques pk_0 et pk_1. Il les donne à $\mathcal{CM}_1$ qui calcule un triplet $\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big)\Big)$ et le donne à $\mathcal{CH}$. Il donne également

$$\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big), (pk_0, pk_1)\Big)$$

à $\mathcal{CM}_2$. Ensuite, $\mathcal{CH}$ choisit :

- $\beta \in \{0, 1\}$ uniformément au hasard,
- M_0 et M_1 dans $\{0, 1\}^k$ uniformément au hasard et
- $S_0 \subseteq \{1, ..., N\}$ uniformément au hasard,

calcule :

- $S_1 := S_0 \oplus \{i_\beta\}$,
- $X_0 := M_0 \oplus \bigoplus_{h \in S_0} f_h$ et $X_1 := M_1 \oplus \bigoplus_{h \in S_1} f_h$ et
- $g := f_{i_\beta} \oplus \kappa \oplus M_0 \oplus M_1$,

invoque :

- $\mathcal{E}_{pk_0}(M_0||S_0)$ pour obtenir c_0 et
- $\mathcal{E}_{pk_1}(M_1||S_1)$ pour obtenir c_1 et

donne $\big((g, X_0, X_1), c_0, c_1\big)$ à $\mathcal{CM}_2$. Vu ce qui lui a été donné par $\mathcal{CM}_1$, $\mathcal{CM}_2$ a à sa disposition

$$\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big), (pk_0, pk_1), (g, X_0, X_1), c_{\mathcal{AS}}, c_0, c_1\Big)$$

pour essayer de deviner β. En notant sa réponse γ, on a

$$Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}(s) = \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|$$

Puis, on observe que l'interaction entre $\mathcal{CH}$ et $\mathcal{CM}$ est également une instance du jeu d'attaque $\mathbf{Exp}^{IND-CPIA-API-MK}_{2,\mathcal{CS},\mathcal{M}_2,\mathcal{PI},\mathcal{CM}}(n)$ où

- $(z_0, z_1, z) = \Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big)\Big)$,
- $e = (pk_0, pk_1)$,
- $(m_{\beta 1}, m_{\beta 2}) = (M_0||S_0, M_1||S_1)$ et
- $y_\beta = (g, X_0, X_1)$.

On peut donc écrire

$$Adv^{IND-CPIA-API-MK}_{2,\mathcal{CS},\mathcal{M}_2,\mathcal{PI},\mathcal{CM}}(s) = \big|\mathbb{P}[\gamma = \beta] - \frac{1}{2}\big|$$

donc finalement

$$Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}} = Adv^{IND-CPIA-API-MK}_{2,\mathcal{CS},\mathcal{M},\mathcal{PI},\mathcal{CM}}$$

et il ne nous reste plus qu'à montrer la négligeabilité du membre de droite.

En accord avec le **principe de simplification,** (voir l'interlude 4.3.2) on peut directement dire que $Adv^{IND-CPIA-API-MK}_{2,\mathcal{CS},\mathcal{M}_2,\mathcal{PICM}}$ est négligeable si aucun PPTA ne peut deviner β à partir de

$$\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big), (g, X_0, X_1)\Big)$$

Or, c'est bien le cas car comme M_0 et M_1 sont choisis uniformément au hasard, le triplet (g, X_0, X_1) est uniformement distribué parmi les triplets de vecteurs binaires de longueur k de somme κ, d'où le lemme. □

Lemme 6 *Si $\mathcal{CM} \in \{\mathcal{DB}_0, \mathcal{DB}_1\}$, l'application*

$$Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$$

est nulle donc négligeable.

Soit $\mathcal{CM}$ l'une des deux bases de données. On peut supposer sans perte de généralité que $\mathcal{CM} = \mathcal{BD}_0$ car $\mathcal{PAM}$ spécifie des rôle symétriques à $\mathcal{DB}_0$ et $\mathcal{DB}_1$.

Maintenant, selon les spécifications de $\mathcal{PAM}$ on peut simuler $\mathbf{Exp}^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$ de la façon suivante. Pour n un paramètre de sécurité, $\mathcal{CM}$ prend en entrée 1^n et calcule un triplet $\Big(i_0, i_1, \big(\kappa, \tau, (i, f_i)_i\big)\Big)$ qu'il donne au challengeur $\mathcal{CH}$. Ce dernier choisit $\beta \in \{0, 1\}$ uniformément au hasard, choisit $S_0 \subseteq \{1, ..., N\}$ uniformément au hasard, choisit $M_0 \in \{0, 1\}^k$ uniformément au hasard et envoie $S_0||M_0$ à $\mathcal{CM}$. L'attaquant $\mathcal{CM}$ dispose alors de

$$\big(i_0, i_1, (i, f_i)_i, S_0||M_0\big)$$

pour déterminer β. Comme S_0 est uniformément distribué et que M_0 ne joue aucun rôle ici, $\mathcal{CM}$ ne peut que deviner β qu'avec une probabilité $\frac{1}{2}$. Autrement dit, $Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$ est identiquement nulle. □

4.3.3 Un protocole avec huit bases de données

Nous allons maintenant traiter le cas d'un protocole opérant dans une infrastructure possédant huit copies de la même base de données. Le but de multiplier les bases de données est de diminuer la quantité de calculs que doit faire le lecteur. Ainsi, dans le cas de deux bases de données ce dernier doit choisir uniformément au hasard un sous ensemble de $\{1, ..., N\}$, où N désigne le nombre d'objets enregistrés dans le système. On peut sans difficulté imaginer qu'en pratique N puisse être de l'ordre de 10^9. Cela implique que $\mathcal{R}$ ait à choisir un élément de $\{0, 1\}^{10^9}$, ce qui est irréaliste.

Composantes structurelles

On se place donc dans le cas où on a huit bases de donneés, chacune d'elles étant une copie de $\mathcal{DB}$. Elles sont indexées par des huit éléments de $\{0, 1\}^3$, i.e.

on les note $\mathcal{DB}_{s_1s_2s_3}$ pour $(s_1, s_2, s_3) \in \{0,1\}^3$, et au lieu de numéroter les flux morphométriques de référence de 1 à N, ils le sont par des triplets (i^1, i^2, i^3) où i^1, i^2 et i^3 sont des entiers parcourant $\{1, ..., \ell\}$, avec $\ell := N^{\frac{1}{3}}$. On peut supposer, quitte à rajouter des faux flux, que ℓ est bien un entier.

Une telle indexation peut s'obtenir simplement en incrémentant les composantes du triplet obtenu en prenant les coefficients du développement de $i \in \{1, .., n\}$ en base ℓ. Les indices obtenus sont alors ordonnés lexicographiquement de $(1, 1, 1)$ à (ℓ, ℓ, ℓ).

Le PIR utilisé dans ce protocole fonctionne presque de la même manière que dans le cas de deux bases de données ; chaque bases envoie à $\mathcal{AS}$ un vecteur binaire, et le XOR de ces huit vecteurs est le flux que l'on veut extraire XORé avec un masque.

On a donc formellement les données suivantes :

- $\{\mathcal{C}\}_{\mathcal{C}} = \{\mathcal{DB}_{s_1s_2s_3}\}_{s_1s_2s_3 \in \{0,1\}^3}$;
- INFRA prend en entrée 1^n, invoque $\mathcal{P}(1^n)$ pour obtenir I, invoque neuf fois de manière indépendante $\mathcal{K}(I)$ pour obtenir neuf couples

$$(pk_{\mathcal{AS}}, sk_{\mathcal{AS}}) \text{ et } (pk_{s_1s_2s_3}, sk_{s_1s_2s_3})_{s_1s_2s_3}$$

donne $sk_{\mathcal{AS}}$ à $\mathcal{AS}$ et $sk_{s_1s_2s_3}$ à $\mathcal{DB}_{s_1s_2s_3}$ et publie toutes les clefs publiques qui sont donc en particulier possedées par $\mathcal{R}$. Puis, il invoque $\mathcal{P}_s(1^n)$ pour obtenir I_s et $\mathcal{K}_s(I_s)$ pour obtenir les clefs de signature (pk_s, sk_s), donne sk_s à $\mathcal{AS}$ et publie pk_s. Le lecteur $\mathcal{R}$ possède ainsi une copie de pk_s. Il invoque $\mathcal{GH}(1^n)$ pour obtenir $\big(\mu, \langle H_\mu \rangle\big)$ et donne ceci à $\mathcal{AS}$ et $\mathcal{R}$. Enfin, les compteurs $C_{\mathcal{AS}}$ et $C_{\mathcal{R}}$ de $\mathcal{AS}$ et $\mathcal{R}$ respectivement sont initialisés à 0.

- ENR prend en entrée une base de données

$$\big(k, N, \tau, (i, f_i)_i\big) = \Big(k, \ell^3, \tau, \big((i^1, i^2, i^3), f_{i^1i^2i^3}\big)_{i^1i^2i^3}\Big)$$

et en donne une copie à chaque $\mathcal{DB}_{s_1s_2s_3}$, et donne ℓ et k à $\mathcal{R}$ et τ à $\mathcal{AS}$.

La ronde d'authentification

L'authentification d'un objet $\mathcal{O}'$ prétendant être $\mathcal{O}$ où $SID_{\mathcal{O}} = i_{\mathcal{O}} = i^1i^2i^3$ est dans la base de données se fait alors comme suit :

1) $\mathcal{R}$ lit $i_{\mathcal{O}}$ dont il obtient $i^1i^2i^3$ et extrait f' ;

2) $\mathcal{R}$ choisit huit masques masques $M_{s_1s_2s_3}$ dans $\{0,1\}^k$ ($s_1s_2s_3$ parcourant $\{0,1\}^3$). Il choisit aussi trois sous-ensembles S_0^1, S_0^2 et S_0^3 de $\{1, ..., \ell\}$ et calcule $S_1 := S_0^1 \oplus \{i^1\}$, $S_1^2 := S_0^2 \oplus \{i^2\}$ et $S_1^3 := S_0^3 \oplus \{i^3\}$. Il calcule ensuite pour tout $s_1s_2s_3$ le chiffrement

$$c_{s_1s_2s_3} := \mathcal{E}_{pk_{s_1s_2s_3}}(S_{s_1}^1 || S_{s_2}^2 || S_{s_3}^3 || M_{s_1s_2s_3})$$

et $c_{\mathcal{AS}} := \mathcal{E}_{pk_{\mathcal{AS}}}(f' \oplus \bigoplus_{s_1s_2s_3} M_{s_1s_2s_3})$. Puis il calcule et garde pour lui-même

$$u := H_\mu(c_{\mathcal{AS}} || \Big|\Big|_{s_1s_2s_3} c_{s_1s_2s_3})$$

et envoie $\big((c_{s_1s_2s_3})_{s_1s_2s_3}, c_{\mathcal{AS}}\big)$ à $\mathcal{AS}$;

3) Pour tout $s_1s_2s_3$, $\mathcal{AS}$ envoie $c_{s_1s_2s_3}$ à $\mathcal{DB}_{s_1s_2s_3}$;

4) Chaque $\mathcal{DB}_{s_2s_2s_3}$ calcule $S^1_{s_1}$, $S^2_{s_2}$ et $S^3_{s_3}$, ainsi que $M_{s_1s_2s_3}$. Puis, il calcule :

$$X_{s_1s_2s_3} := M_{s_1s_2s_3} \oplus \bigoplus_{h^1h^2h^3 \in S^1_{s_1} \times S^2_{s_2} \times S^3_{s_3}} f_{h^1h^2h^3}$$

Ensuite, $\mathcal{DB}_{s_1s_2s_3}$ envoie $X_{s_1s_2s_3}$ à $\mathcal{AS}$;

5) $\mathcal{AS}$ calcule $f' \oplus \bigoplus_{s_1s_2s_3} M_{s_1s_2s_3}$ et la somme

$$(f' \oplus \bigoplus_{s_1s_2s_3} M_{s_1s_2s_3}) \oplus \bigoplus_{s_1s_2s_3} X_{s_1s_2s_3}$$

pour obtenir $f' \oplus f_{i_{\mathcal{O}}}$. Il détermine le poids w de Hamming du vecteur obtenu et retourne la réponse A selon que $w \leq \tau$ ou $w > \tau$. Il calcule $u := H_\mu(c_{\mathcal{AS}} || \|_{s_1s_2s_3} c_{s_1s_2s_3})$, invoque $Sign_{sk_s}(A, u)$ pour obtenir une signature digitale $\mathfrak{s}$ sur (A, u) et envoie $\big((A, u), \mathfrak{s}\big)$ à $\mathcal{R}$.

6) $\mathcal{R}$ regarde si le u qu'il a gardé en mémoire est égal à celui envoyé par $\mathcal{AS}$. Il vérifie ensuite la signature digitale $\mathfrak{s}$ sur $(A, u, \mathfrak{s})$. Si l'une de ces vérifications échoue, $\mathcal{R}$ retourne $\perp$ et invalide la ronde. Sinon, il accepte A.

Coût en calculs

Explicitons le travail des composantes lors d'une ronde d'authentification.

• Le lecteur $\mathcal{R}$ doit générer de manière pseudo-aléatoire huit vecteurs binaires de longueur k et trois vecteur binaires de longueur $\ell = N^{\frac{1}{3}}$. Il doit chiffrer un message de longueur k, et huit messages de longueur 3ℓ. Il doit maintenir un compteur $C_{\mathcal{R}}$. Si $lg(c_k)$ et $lg(c_{k+\ell})$ désignent la longueur des chiffrés de messages de longueur k et $k + \ell$ respectivement, il doit calculer le hachage d'un message de longueur $lg(c_k) + 8lg(c_{k+\ell})$. Enfin, il doit vérifier une signature digitale.

Au total, le lecteur doit choisir uniformément au hasard $8k + 3N^{\frac{1}{3}}$ bits en plus de l'aléa nécessaire au chiffrement de huit messages de longueur $k + \ell$ et d'un message de longueur k. Il envoie au serveur d'authentification un message de longueur $lg(c_k) + 8lg(c_{k+\ell})$.

• Le serveur d'authentification $\mathcal{AS}$ doit déchiffrer un message de longueur $lg(c_k)$, calculer le hachage d'un message de longueur $lg(c_k) + 8lg(c_{k+\ell})$, maintenir un compteur $C_{\mathcal{AS}}$ et calculer une signature digitale.

• Les bases de données doivent chacune déchiffrer un message de longueur $lg(c_\ell)$ et calculer le XOR d'au plus N flux morphométriques de longueur k.

Adversaires extérieurs

Comme dans le paragraphe 4.3.2, on regarde les adversaires extérieurs en premier.

Intégrité

Théorème 12 *Si le schéma de signature $\mathcal{DS}$ a des signatures inforgeables et la famille de fonctions de hachage $\mathfrak{H}$ résiste aux collisions, le schéma d'authentification $\mathcal{PAM}$ garantit l'intégrité de la réponse d'une ronde d'authentification contre les adversaires extérieurs.*

Soient comme d'habitude n un paramètre de sécurité et $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4)$ un PPTA jouant contre $\mathcal{PAM}$ dans le jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{B}}(n)$. Il s'agit de construire $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ utilisant $\mathcal{B}$ comme sous-programme pour soit trouver une collision pour $\mathfrak{H}$, soit forger une signature pour $\mathcal{DS}$ dans une simulation avec un challengeur $\mathcal{CH}$.

- *Phase de mise en place par $\mathcal{CH}$ des paramètres pour $\mathcal{A}$*

$\mathcal{CH}$ invoque $\mathcal{P}_s(1^n)$ pour obtenir I_s et $\mathcal{K}_s(I_s)$ pour obtenir les clefs (pk_s, sk_s) du schéma de signature. Il invoque $\mathcal{GH}(1^n)$ pour obtenir une instance de fonction de hachage $\big(\mu, \langle H_\mu \rangle\big)$. Il envoie $\big(pk_s, \mu H_\mu \rangle\big)$ à $\mathcal{A}_1$.

- *Phase de mise en place par $\mathcal{A}$ des paramètres pour $\mathcal{B}$*

$\mathcal{A}_1$ invoque une fois $\mathcal{P}(1^n)$ pour obtenir I et neuf fois $\mathcal{K}(I)$ pour calculer les clefs de chiffrement $\big((pk_{\mathcal{AS}}, sk_{\mathcal{AS}}), (pk_{s_1s_2s_3}, sk_{s_1s_2s_3})_{s_1s_2s_3}\big)$.

- *Phase d'appels à l'oracle de signature par $\mathcal{A}$*

L'attaquant $\mathcal{A}_1$ répond aux requêtes de $\mathcal{B}_1$ en les transmettant comme requêtes à $\mathcal{CH}$. Le PPTA $\mathcal{A}$ invoque $\mathcal{B}_1\big(pk_s, pk_{\mathcal{AS}}, (pk_{s_1s_2s_3})_{s_1s_2s_3}, \langle H_\mu \rangle\big)$ et opère comme suit.

Lorsque $\mathcal{B}_1$ spécifie une o-ième requête de la forme

$$\Big(j_o^1 j_o^2 j_o^3, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big),$$

$\mathcal{A}_1$ simule une ronde d'authentification valide avec comme entrées la base de données $\big(k_o, N_o, \tau_o, (i, f_{oi})_i\big)$, le flux de référence $f_{oj_o^1 j_o^2 j_o^3}$ et le flux extrait $f_{oj_o^1 j_o^2 j_o^3} \oplus \kappa_o$. Précisément, $\mathcal{A}_1$ choisit des masques $M_{os_1s_2s_3}$ dans $\{0,1\}^{k_o}$ uniformément au hasard, choisit S^1_{o0}, S^2_{o0} et S^3_{o0} dans l'ensemble des parties de $\{1, ..., \ell_o\}$ uniformément au hasard, calcule $S^1_{o1} := S^1_{o0} \oplus \{j_o^1\}$, $S^2_{o1} := S^2_{o0} \oplus \{j_o^2\}$ et $S^3_{o1} := S^3_{o0} \oplus \{j_o^3\}$, calcule

$$g_o := f_{oj_o^1 j_o^2 j_o^3} \oplus \kappa_o \oplus \bigoplus_{s_1s_2s_3} M_{os_1s_2s_3},$$

invoque $\mathcal{E}_{pk_{\mathcal{AS}}}(g_o)$ pour obtenir $c_{o\mathcal{AS}}$, invoque $\mathcal{E}_{pk_{s_1s_2s_3}}(M_{os_1s_2s_3}||S^1_{os_1}||S^2_{os_2}||S^3_{os_3})$ pour obtenir $c_{os_1s_2s_3}$ pour tout $s_1s_2s_3$, incrémente le compteur du lecteur $C_{\mathcal{R}}$ qui vaut maintenant o, pose

$$c_o = c_{o\mathcal{AS}}|| \bigg|\bigg| \limits_{os_1s_2s_3} c_{s_1s_2s_3}$$

et calcule $u_o := H_\mu(c_o||o)$. Puis il détermine la réponse A_o en fonction de κ_o et τ_o, incrémente le compteur du serveur d'authentification $C_{\mathcal{AS}}$ (valant maintenant aussi o) et envoie (A_o, u_o) comme requête de signature à $\mathcal{CH}$ qui invoque alors $Sign_{sk_s}(A_o, u_o)$ pour obtenir $\mathfrak{s}_o$ et le retourne à $\mathcal{A}_1$. Ce dernier peut à présent répondre à $\mathcal{B}_1$ en lui fournissant $\Big(o, c_o, \big((A_o, u_o), \mathfrak{s}_o\big)\Big)$.

Quand $\mathcal{B}_1$ cesse de faire des requêtes, $\mathcal{A}_1$ et $\mathcal{B}_1$ possèdent chacun les données :

$$\mathfrak{Q} := \left(\Big(j^1_o j^2_o j^3_o, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big), \Big(o, c_o, \big(\big(j(\kappa_o, \tau_o), u_o\big), \mathfrak{s}_o\big)\Big)\right)_o$$

pour o parcourant $\{1, ..., q\}$ et q étant majoré par un polynôme en n. La première partie d'une telle donnée est la requête et la deuxième le résultat où la réponse finale A_o vaut $j(\kappa_o, \tau_o)$.

Comme dans les cas précédents, ces requêtes peuvent ne pas être deux-à-deux distinctes.

Enfin, $\mathcal{B}_1$ retourne la liste $\mathfrak{Q}$ ci-dessus et une information d'état e_1.

- *Requête de challenge de $\mathcal{B}$*

  $\mathcal{A}_1$ invoque $\mathcal{B}_2(\mathfrak{Q}, e_1)$ pour obtenir une requête de challenge

$$\Big(j^1 j^2 j^3, \big(k, N, \tau, (i, f_i)_i\big), \kappa, e_2\Big)$$

qui peut déjà apparaître dans la liste des requêtes.

- *Première phase de réponse au challenge : $\mathcal{A}$ simule le premier message de $\mathcal{R}$*

  $\mathcal{A}_1$ choisit les huit masques $M_{s_1s_2s_3}$ uniformément au hasard dans $\{0,1\}^k$, choisit S^1_0, S^2_0 et S^3_0 uniformément au hasard parmi les sous-ensembles de $\{1, ..., \ell\}$, calcule $S^1_1 := S^1_0 \oplus \{j^1\}$, $S^2_1 := S^2_0 \oplus \{j^2\}$ et $S^3_1 := S^3_0 \oplus \{j^3\}$, calcule

$$g := f_{j^1 j^2 j^3} \oplus \kappa \oplus \bigoplus_{s_1s_2s_3} M_{s_1s_2s_3},$$

invoque $\mathcal{E}_{pk_{\mathcal{AS}}}(g)$ et les $\mathcal{E}_{pk_{s_1s_2s_3}}(M_{s_1s_2s_3}||S^1_{s_1}||S^2_{s_2}||S^3_{s_3})$ pour obtenir respectivement $c_{\mathcal{AS}}$ et les $c_{s_1s_2s_3}$, incrémente $C_{\mathcal{R}}$ qui vaut donc $q+1$, pose

$$c = c_{\mathcal{AS}}|| \bigg|\bigg| \limits_{s_1s_2s_3} c_{s_1s_2s_3}$$

et calcule $u := H_\mu\big(c||(q+1)\big)$.

• *Deuxième phase de réponse au challenge : $\mathcal{B}$ manipule les messages de $\mathcal{R}$ vers $\mathcal{AS}$*

$\mathcal{A}_1$ invoque $\mathcal{B}_2\big(c_{\mathcal{AS}}, (c_{s_1s_2s_3})_{s_1s_2s_3}, e_1\big)$ pour obtenir $\big(c'_{\mathcal{AS}}, (c'_{s_1s_2s_3})_{s_1s_2s_3}, e_2\big)$ de sorte qu'il existe $g' \in \{0,1\}^k$ et $L_{s_1s_2s_3}$ dans $\{0,1\}^{k+3\ell}$ pour tout $s_1s_2s_3$ vérifiant $c'_{\mathcal{AS}} \in \mathcal{E}_{pk_{\mathcal{AS}}}(g')$ et $c'_{s_1s_2s_3} \in \mathcal{E}_{pk_{s_1s_2s_3}}(L_{s_1s_2s_3})$. On fait ainsi en sorte que les calculs qui suivent aient un sens.

• *Troisième phase de réponse au challenge : $\mathcal{A}$ simule les calculs de l'interaction entre $\mathcal{AS}$ et les composantes $\mathcal{DB}_0$ et $\mathcal{DB}_1$*

$\mathcal{A}_1$ calcule g' et les $L_{s_1s_2s_3}$ avec les clefs secrètes et partitionne chaque $L_{s_1s_2s_3}$ en

$$M'_{s_1s_2s_3}||{S^1_{s_1}}'||{S^2_{s_2}}'||{S^3_{s_3}}'$$

où les $M'_{s_1s_2s_3}$ sont dans $\{0,1\}^k$ et les ${S^i_{s_i}}'$ sont dans $\{0,1\}^\ell$. Ensuite, $\mathcal{A}_1$ calcule

$$X'_{s_1s_2s_3} := M'_{s_1s_2s_3} \oplus \bigoplus_{h^1h^2h^3 \in {S^1_{s_1}}' \times {S^2_{s_2}}' \times {S^3_{s_3}}'} f_{h^1h^2h^3},$$

calcule $\kappa' := g' \oplus \bigoplus_{s_1s_2s_3} X'_{s_1s_2s_3}$, calcule le poids de Hamming de κ', calcule A' en fonction de ce poids, incrémente le compteur $C_{\mathcal{AS}}$ de $\mathcal{AS}$ qui vaut donc maintenant $(q+1)$ et calcule

$$u' := H_\mu\big(c'_{\mathcal{AS}}|| \bigm|\bigm| _{s_1s_2s_3} c'_{s_1s_2s_3}||(q+1)\big)$$

Puis, il envoie (A', u') à $\mathcal{CH}$.

• *Quatrième phase de réponse au challenge : $\mathcal{A}$ obtient de $\mathcal{CH}$ une signature digitale pour simuler le message de $\mathcal{AS}$ à $\mathcal{R}$*

$\mathcal{CH}$ invoque $Sign_{sk_s}(A', u')$ pour obtenir $\mathfrak{s}'$ et envoie $\mathfrak{s}'$ à $\mathcal{A}_2$.

• *Réponse au challenge de $\mathcal{B}$: $\mathcal{B}$ manipule les messages de $\mathcal{AS}$ vers $\mathcal{R}$*

$\mathcal{A}_2$ invoque $\mathcal{B}_4\big((A', u'), \mathfrak{s}', e_3\big)$ pour obtenir $\big((A'', u''), \mathfrak{s}''\big)$ où $A'' \in \{0,1\}$.

• *Réponse au challenge de $\mathcal{A}$: $\mathcal{A}$ cherche dans ce qu'il a calculé une collision ou une signature forgée*

Si $A'' = j(\kappa, \tau) \oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'', u''), \mathfrak{s}''\big) = 1$, $\mathcal{A}_2$ pose $cm_{\mathcal{R}_2} = A''$. Puis :

- si $(A'', u'') = (A', u')$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c'||(q+1)\big)$ et
- si $(A'', u'') \neq (A', u')$:
  - si $(A'', u'') \notin \big\{(A_o, u_o)\big\}_o$, $\mathcal{A}$ retourne (A'', u'') et
  - s'il existe o tel que $(A'', u'') = (A_o, u_o)$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c_o||o\big)$.

Dans tous les autres cas, $\mathcal{A}_2$ retourne un message d'échec.

Etudions la relation entre les avantages. Les conditions $A'' = j(\kappa,\tau) \oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'',u''),s''\big) = 1$ sont simultanément satisfaites si et seulement si $cm_{\mathcal{R}_2} = j(\kappa,\tau) \oplus 1$ à la fin du jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{B}}(n)$. Ensuite, que ces conditions soient satisfaites entraîne soit $c' \neq c$ soit $(A'',u'') \neq (A',u')$ faute de quoi $\mathcal{B}$ n'attaque pas le challenge et la réponse finale de l'authentification est $j(\kappa,\tau)$. Il vient donc que

• si $(A'',u'') = (A',u')$, $c' \neq c$ et $\big(c'||(q+1), c||(q+1)\big)$ est une collision pour H_μ car $u'' = u$ et $u'' = u'$ donc

$$H_\mu\big(c'||(q+1)\big) = u' = u = H_\mu\big(c||(q+1)\big)$$

et

• si $(A'',u'') \neq (A',u')$, que $Verify_{pk_s}\big((A'',u''),\mathfrak{s}''\big) = 1$ implique que $\mathfrak{s}''$ est une signature valide sur le message (A'',u'') et il reste à voir si (A'',u'') a déjà fait l'objet d'une requête de signature ou non.

Si $(A'',u'') \neq (A_o,u_o)$ pour tout o, (A'',u'') ne fait pas partie des signatures demandées à $\mathcal{CH}$ et donc c'est une vraie signature forgée.

S'il existe o tel que $(A'',u'') = (A_o,u_o)$, on doit avoir $c||(q+1) \neq c_o||o$ car $o \leq q$ et on obtient une collision pour H_μ étant donné que

$$H_\mu(c_o||o) = u_o = u'' = u = H_\mu\big(c||(q+1)\big)$$

Il est clair que $\mathcal{A}$ joue contre $\mathfrak{H}$ et $\mathcal{DS}$ dans le jeu $\mathbf{Exp}^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}}$ donc

$$Adv^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}} \geq Adv^{int-ext}_{\mathcal{PAM},\mathcal{B}}$$

ce qui permet de conclure. □

Confidentialité

Théorème 13 *Si le schéma de chiffrement $\mathcal{CS}$ est ind-cpa-sécurisé, le schéma d'authentification $\mathcal{PAM}$ garantit l'anonymat d'objet contre les adversaires extérieurs.*

Soit $\mathcal{A} := (\mathcal{A}_1,\mathcal{A}_2)$ un PPTA jouant contre $\mathcal{PAM}$ dans le jeu $\mathbf{Exp}^{oa-ext}_{\mathcal{PAM},\mathcal{A}}$. Pour n un paramètre de sécurité et $\mathcal{CH}$ un challengeur, on décrit le déroulement de ce jeu comme suit.

Le challengeur $\mathcal{CH}$ invoque une fois $\mathcal{P}(1^n)$ pour obtenir I puis neuf fois $\mathcal{K}(I)$ pour obtenir les clefs publiques $(pk_{\mathcal{AS}}, pk_{s_1s_2s_3})$ pour $s_1s_1s_3 \in \{0,1\}^3$ qu'il donne à $\mathcal{A}_1$. Ce dernier génère $\Big(i_0,i_1,\big(k,N,\tau,(i,f_i)_i\big),\kappa\Big)$ qu'il donne à $\mathcal{CH}$, et donne

$$\Big(i_0,i_1,\big(k,N,\tau,(i,f_i)_i\big),\kappa,\big(pk_{\mathcal{AS}},(pk_{s_1s_2s_3})_{s_1s_2s_3}\big)\Big)$$

à $\mathcal{A}_2$.

Le challengeur $\mathcal{CH}$ choisit :

- $\beta \in \{0,1\}$ uniformément au hasard,
- $M_{s_1s_2s_3} \in \{0,1\}^k$ pour tout $s_1s_2s_3$ uniformément au hasard et
- S_0^1, S_0^2 et S_0^3 dans $\{1,...,\ell\}$ uniformément au hasard,

calcule :

- $g := f_{i_\beta} \oplus \kappa \oplus \bigoplus_{s_1s_2s_3} M_{s_1s_2s_3}$,
- $S_1^1 := S_0^1 \oplus \{i_\beta^1\}$, $S_1^2 := S_0^2 \oplus \{i_\beta^2\}$ et $S_1^3 := S_0^3 \oplus \{i_\beta^3\}$ et
- $X_{s_1s_2s_3} := M_{s_1s_2s_3} \oplus \bigoplus_{h^1h^2h^3 \in S_{s_1}^1 \times S_{s_2}^2 \times S_{s_3}^3} f_{h^1h^2h^3}$ pour tout $s_1s_2s_3$ et

invoque

- $\mathcal{E}_{pk_{\mathcal{AS}}}(g)$ pour obtenir $c_{\mathcal{AS}}$,
- $\mathcal{E}_{pk_{s_1s_2s_3}}(S_{s_1}^1||S_{s_2}^2||S_{s_3}^3||M_{s_1s_2s_3})$ pour obtenir $c_{s_1s_2s_3}$ pour tout $s_1s_2s_3$ et

donne $\Big((X_{s_1s_2s_3})_{s_1s_2s_3}, \big(c_{\mathcal{AS}}, (c_{s_1s_2s_3})_{s_1s_2s_3}\big)\Big)$ à $\mathcal{A}_2$.

Pour essayer de calculer β, $\mathcal{A}_2$ possède donc

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, \big((X_{s_1s_2s_3})_{s_1s_2s_3}\big), \big(pk_{\mathcal{AS}}, (pk_{s_1s_2s_3})_{s_1s_2s_3}\big), \big(c_{\mathcal{AS}}, (c_{s_1s_2s_3})_{s_1s_2s_3}\big)\Big)$$

et l'interaction précédente est clairement une instance du jeu $\mathbf{Exp}^{ind-cpia-api-mk}_{9,\mathcal{CS},\mathcal{M}_9,\mathcal{PI},\mathcal{A}}(n)$ donc comme $\mathcal{CS}$ est *ind-cpa*-sécurisé la proposition 6 et le principe de simplification impliquent que si tout PPTA ne peut deviner β à partir de

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, (X_{s_1s_2s_3})_{s_1s_2s_3}\Big)$$

qu'avec une probabilité $\frac{1}{2}$, l'application $Adv^{ind-cpia-api-mk}_{9,\mathcal{CS},\mathcal{M}_9,\mathcal{PI},\mathcal{CM}}$ est négligeable. Les masques $M_{s_1s_2s_3}$ étant choisis uniformément au hasard, les $X_{s_1s_2s_3}$ sont uniformément distribués ce qui permet de conclure. □

Composantes malicieuses

Théorème 14 *Si le schéma de chiffrement $\mathcal{CS}$ est ind-cpa-sécurisé, le schéma d'authentification $\mathcal{PAM}$ garantit l'anonymat d'objet contre les composantes serveur malicieuses.*

Comme avec deux bases de données, les deux lemmes suivants permettent de conclure.

Lemme 7 *Si $\mathcal{CS}$ est ind-cpa-sécurisé et $\mathcal{CM} = \mathcal{AS}$, l'application*

$$Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$$

est négligeable.

On simule une instance de $\mathbf{Exp}^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$ avec $\mathcal{CM} = \mathcal{AS}$ et $\mathcal{CH}$ un challengeur.

Pour $n \in \mathbb{N}$ un paramètre de sécurité, $\mathcal{CH}$ invoque une fois $\mathcal{P}(1^n)$ pour obtenir I et huit fois $\mathcal{K}(I)$ pour obtenir les clefs publiques $pk_{s_1s_2s_3}$ pour $s_1s_1s_3 \in \{0,1\}^3$. Il donne alors ces clefs à $\mathcal{CM}_1$. Ce dernier génère $\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa\Big)$ qu'il donne à $\mathcal{CH}$, et donne

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, (pk_{s_1s_2s_3})_{s_1s_2s_3}\Big)$$

à $\mathcal{CM}_2$.

Le challengeur $\mathcal{CH}$ choisit :

- $\beta \in \{0,1\}$ uniformément au hasard,
- $M_{s_1s_2s_3} \in \{0,1\}^k$ pour tout $s_1s_2s_3$ uniformément au hasard et
- S^1_0, S^2_0 et S^3_0 dans $\{1, ..., \ell\}$ uniformément au hasard,

calcule :

- $g := f_{i_\beta} \oplus \kappa \oplus \bigoplus_{s_1s_2s_3} M_{s_1s_2s_3}$,
- $S^1_1 := S^1_0 \oplus \{i^1_\beta\}$, $S^2_1 := S^2_0 \oplus \{i^2_\beta\}$ et $S^3_1 := S^3_0 \oplus \{i^3_\beta\}$ et
- $X_{s_1s_2s_3} := M_{s_1s_2s_3} \oplus \bigoplus_{h^1h^2h^3 \in S^1_{s_1} \times S^2_{s_2} \times S^3_{s_3}} f_{h^1h^2h^3}$ pour tout $s_1s_2s_3$ et

invoque

- $\mathcal{E}_{pk_{s_1s_2s_3}}(S^1_{s_1}||S^2_{s_2}||S^3_{s_3}||M_{s_1s_2s_3})$ pour obtenir $c_{s_1s_2s_3}$ pour tout $s_1s_2s_3$ et

donne $\big(g, (X_{s_1s_2s_3})_{s_1s_2s_3}, (c_{s_1s_2s_3})_{s_1s_2s_3}\big)$ à $\mathcal{A}_2$.

Ainsi, pour tenter de déterminer β, $\mathcal{A}_2$ possède

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, \big(g, (X_{s_1s_2s_3})_{s_1s_2s_3}\big), (pk_{s_1s_2s_3})_{s_1s_2s_3}, (c_{s_1s_2s_3})_{s_1s_2s_3}\Big)$$

et cette interaction est une instance de $\mathbf{Exp}^{ind-cpia-api-mk}_{8,\mathcal{CS},\mathcal{M}_8,\mathcal{PI},\mathcal{CM}}(n)$ donc comme $\mathcal{CS}$ est *ind-cpa*-sécurisé la proposition 6 (avec le principe de simplification) entraîne que si tout PPTA ne peut deviner β à partir de

$$\Big(i_0, i_1, \big(k, N, (i, f_i)_i\big), \kappa, \big(g, (X_{s_1s_2s_3})_{s_1s_2s_3}\big)\Big)$$

qu'avec une probabilité $\frac{1}{2}$, l'application $Adv^{ind-cpia-api-mk}_{8,\mathcal{CS},\mathcal{M}_8,\mathcal{PI},\mathcal{CM}}$ est négligeable. Les masques $M_{s_1s_2s_3}$ étant choisis uniformément au hasard, le vecteur

$$\big(g, (X_{s_1s_2s_3})_{s_1s_2s_3}\big)$$

est uniformément distribué parmi les 9-uplets d'éléments de $\{0,1\}^k$ de somme κ, et on a bien ce qu'on veut. □

Lemme 8 *Si $\mathcal{CM} \in \{\mathcal{DB}_{s_1s_2s_3}\}_{s_1s_2s_3}$, l'application*

$$Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$$

est nulle donc négligeable.

Soit $\mathcal{CM}$ l'une des huit bases de données, qu'on peut supposer être $\mathcal{CM} = \mathcal{BD}_{000}$ pour des raisons de symétrie.

On simule alors $\mathbf{Exp}^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}(n)$ pour un $n \in \mathbb{N}$: $\mathcal{CM}$ prend en entrée 1^s et calcule $\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa\Big)$ qu'il donne au challengeur $\mathcal{CH}$. Ce dernier choisit $\beta \in \{0,1\}$ uniformément au hasard, choisit S_0^1, S_0^2 et S_0^3 dans $\{1, ..., \ell\}$, uniformément au hasard, choisit $M_{000} \in \{0,1\}^k$ uniformément au hasard et envoie $S_0^1||S_0^2||S_0^3||M_0$ à $\mathcal{CM}$. Le PPTA $\mathcal{CM}$ possède donc

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, S_0^1||S_0^2||S_0^3||M_{000}\Big)$$

pour déterminer β. Vu la distribution uniforme de S_0^1, S_0^2 et S_0^3, la probabilité que $\mathcal{CM}$ devine β est $\frac{1}{2}$, i.e. $Adv^{oa-comp}_{\mathcal{PAM},\mathcal{CM}}$ est nulle. □

4.3.4 Des protocoles où deux bases de données font le travail de huit

Nous avons précisé au début de la section 4.3.3 que le principal défaut du protocole à deux bases de données opérant à l'aide du PIR le plus simple est son coût en calculs pour le lecteur. Pour remédier à cette situation, nous avons augmenté le nombre de bases de données afin d'utiliser un PIR légèrement plus sophistiqué permettant de rendre ces calculs réalistes. Par contre, il faut maintenant gérer un système avec huit bases de données identiques, qui ne doivent pas entrer en collusion, et possèdant chacune sa propre paire clef secrète/clef publique pour le schéma de chiffrement $\mathcal{CS}$. Si en théorie les choses marchent bien, l'implémentation du système reste quand même difficile.

C'est la raison d'être des deux protocoles que nous présentons ici. Tous deux utilisent un PIR permettant d'utiliser deux bases de données seulement pour en simuler huit. Ils diffèrent en efficacité : le premier des deux, appelé "semi-final" et noté $\mathcal{PAM}^{semi-final}$, demande au lecteur de choisir deux masques de taille $k\ell$ alors que le deuxième, appelé "final" et noté $\mathcal{PAM}^{final}$, demande au lecteur de choisir huit masques de taille k. L'avantage est clairement du côté du deuxième protocole. Malheureusement, les résultats de sécurité théorique dont nous disposons ne nous ont pas encore permi de conclure que le protocole final est sécurisé comme souhaité au niveau du serveur d'authentification. Par contre, une telle preuve existe pour le protocole semi-final.

Dans la suite, nous donnons la description des deux protocoles, et les preuves de sécurité que nous avons pu établir pour chacun d'eux. Les deux protocoles ont les mêmes composantes structurelles et les fonctionnalités Infra et Enr sont identiques pour les deux également. Ils ne diffèrent que par la ronde d'authentification.

4.3.4.1 Composantes structurelles

La structure fondamentale de ces protocoles est donnée par :

• $\{\mathcal{C}\}_{\mathcal{C}} = \{\mathcal{DB}_0, \mathcal{DB}_1\}$: on est revenu à deux bases de données, et elles doivent faire le travail de huit d'entre elles ;

• INFRA prend en entrée 1^n, invoque $\mathcal{P}(1^n)$ pour obtenir I et invoque trois fois de manière indépendante $\mathcal{K}(I)$ pour obtenir les couples $(pk_{\mathcal{AS}}, sk_{\mathcal{AS}})$, (pk_0, sk_0) et (pk_1, sk_1), donne $sk_{\mathcal{AS}}$ à $\mathcal{AS}$, sk_0 à $\mathcal{DB}_0$ et sk_1 à $\mathcal{DB}_1$ et publie les clefs publiques que donc $\mathcal{R}$ récupère. Il invoque $\mathcal{P}_s(1^n)$ pour avoir I_s, $\mathcal{K}_s(I_s)$ pour obtenir les clefs de signature (pk_s, sk_s), donne sk_s à $\mathcal{AS}$ et publie pk_s. Le lecteur $\mathcal{R}$ possède ainsi une copie de pk_s. Il invoque $\mathcal{GH}(1^n)$ pour obtenir $\big(\mu, \langle H_\mu \rangle\big)$ et donne ceci à $\mathcal{AS}$ et $\mathcal{R}$. Enfin, les compteurs $C_{\mathcal{AS}}$ et $C_{\mathcal{R}}$ de $\mathcal{AS}$ et $\mathcal{R}$ respectivement sont initialisés à 0 ;

• ENR prend en entrée une base de données

$$\big(k, N, \tau, (i, f_i)_i\big) = \Big(k, \ell^3, \tau, \big((i_1, i_2, i_3), f_{i_1 i_2 i_3}\big)_{i_1 i_2 i_3}\Big)$$

et en donne une copie à $\mathcal{DB}_0$ et $\mathcal{DB}_1$.

4.3.4.2 Un protocole semi-final : ronde d'authentification

Pour authentifier un objet $\mathcal{O}'$ prétendant être $\mathcal{O}$ enregistré à l'indice $i_{\mathcal{O}}$ dans le système, on procède comme suit :

1) $\mathcal{R}$ lit $i_{\mathcal{O}}$ et extrait f' ;

2) $\mathcal{R}$ entame le PIR (deux bases de données en simulant huit) :

a) $\mathcal{R}$ obtient (i^1, i^2, i^3) de $i_{\mathcal{O}}$;

b) $\mathcal{R}$ choisit S_0^1, S_0^2, et S_0^3 dans $\{1, ..., \ell\}$ uniformément au hasard ;

c) $\mathcal{R}$ calcule $S_1^1 := S_0^1 \oplus \{i^1\}$, $S_1^2 := S_0^2 \oplus \{i^2\}$, et $S_1^3 := S_0^3 \oplus \{i^3\}$;

d) $\mathcal{R}$ choisit deux (gros) masques M_0 et M_1 dans $\{0,1\}^{k(3\ell+1)}$ uniformément au hasard ;

e) $\mathcal{R}$ choisit $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard ;

f) $\mathcal{R}$ partitionne M_0 et M_1 en $3\ell + 1$ blocs de longueur k :

$$M_0 = M_0(1)||...||M_0(3\ell + 1)$$

et

$$M_1 = M_1(1)||...||M_1(3\ell + 1)$$

g) $\mathcal{R}$ calcule

$$c_0 := \mathcal{E}_{pk_0}(S_0^1||S_0^2||S_0^3||M_0||\sigma),$$
$$c_1 := \mathcal{E}_{pk_1}(S_1^1||S_1^2||S_1^3||M_1||\sigma)$$

et

$$c_{\mathcal{AS}} := \mathcal{E}_{pk_{\mathcal{AS}}}\big(f' \oplus M||\sigma(i^1)||\sigma(\ell + i^2)||\sigma(2\ell + i^3)\big),$$

où

$$M := \bigoplus_{b \in \{0,1\}} M_b(i^1) \oplus M_b(\ell + i^2) \oplus M_b(2\ell + i^3) \oplus M_b(3\ell + 1)$$

$\mathcal{R}$ calcule et conserve $u := H_\mu(c_{\mathcal{AS}}||c_0||c_1)$ et envoie $(c_0, c_1, c_{\mathcal{AS}})$ à $\mathcal{AS}$;

3) $\mathcal{AS}$ transmet c_0 à $\mathcal{DB}_0$ et c_1 à $\mathcal{DB}_1$;

4) $\mathcal{DB}_0$ (resp., $\mathcal{DB}_1$) calcule

$$S_0^1||S_0^2||S_0^3||M_0||\sigma$$

(resp.,

$$S_1^1||S_1^2||S_1^3||M_1||\sigma),$$

puis il calcule :

a)

$$X_0'' := X_{100}(1)||...||X_{100}(\ell)||X_{010}(1)||...||X_{010}(\ell)||X_{001}(1)||...||X_{001}(\ell)||X_{000}$$

(resp.,

$$X_1'' := X_{011}(1)||...||X_{011}(\ell)||X_{101}(1)||...||X_{101}(\ell)||X_{110}(1)||...||X_{110}(\ell)||X_{111});$$

b)

$$X_0' := M_0 \oplus X_0''$$

(resp.,

$$X_1' := M_1 \oplus X_1'');$$

On rappelle sur deux exemples les définitions des sommes de la forme $X_{100}(t)$, $X_{010}(t)$,...,$X_{110}(t)$. On a :

$$X_{100}(t) = \bigoplus_{S_0^1 \oplus \{t\} \times S_0^2 \times S_0^3} f_{h^1 h^2 h^3}$$

et

$$X_{110}(t) = \bigoplus_{S_1^1 \times S_1^2 \times S_1^3 \oplus \{t\}} f_{h^1 h^2 h^3}$$

c) $X_0 := X_0'$ auquel σ est appliquée aux 3ℓ premiers blocs de longueur k (de même pour X_1 avec X_1').

Les vecteurs X_0 et X_1 sont tous deux dans $\{0,1\}^{k(3\ell+1)}$. Pour tout $t \in \{1, ..., 3\ell+1\}$, on note $X_0(t)$ (resp., $X_1(t)$) le t-ième bloc en partant de la gauche de longueur k de X_0 (resp., X_1). Ainsi, on a

$$X_0 = X_0(1)||...||X_0(3\ell+1)$$

et

$$X_1 = X_1(1)||...||X_1(3\ell+1)$$

$\mathcal{DB}_0$ (resp., $\mathcal{DB}_1$) envoie X_0 (resp., X_1) à $\mathcal{AS}$;

5) $\mathcal{AS}$ calcule $g := f' \oplus M$, $\sigma(i^1)$,$\sigma(\ell + i^2)$, et $\sigma(2\ell + i^3)$, et calcule

$$g \oplus \bigoplus_{b\in\{0,1\}} X_b\big(\sigma(i^1)\big) \oplus X_b\big(\sigma(\ell + i^2)\big) \oplus X_b\big(\sigma(2\ell + i^3)\big) \oplus X_b(3\ell + 1)$$

qui n'est autre que $f_{i_{\mathcal{O}}} \oplus f'$. Ensuite, $\mathcal{AS}$ calcule le poids de Hamming de $f_{i_{\mathcal{O}}} \oplus f'$ et détermine sa réponse A en fonction de la comparaison du poids avec τ. Enfin, il calcule $u := H_\mu(c_{\mathcal{AS}}||c_0||c_1)$, invoque $Sign_{sk_s}(A, u)$ pour obtenir une signature digitale $\mathfrak{s}$ sur (A, u) et envoie $\big((A, u), \mathfrak{s}\big)$ à $\mathcal{R}$.

6) $\mathcal{R}$ regarde si le u qu'il a gardé en mémoire est égal à celui envoyé par $\mathcal{AS}$. Il vérifie ensuite la signature digitale $\mathfrak{s}$ sur $\big((A, u), \mathfrak{s}\big)$. Si l'une de ces vérifications échoue, $\mathcal{R}$ retourne $\perp$ et invalide la ronde. Sinon, il accepte A.

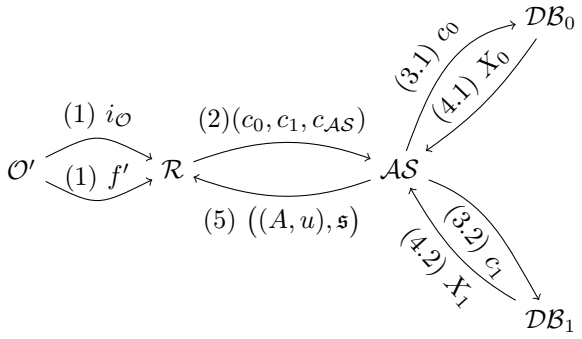


FIGURE 4.2 – Le protocole semi-final

4.3.4.3 Coût en calculs

Evaluons le travail des composantes individuelles.

• Le lecteur $\mathcal{R}$ doit choisir de manière pseudo-aléatoire deux vecteurs binaires de longueur $k(3\ell + 1)$ (où $\ell = N^{\frac{1}{3}}$), trois vecteurs binaires de longueur ℓ, et une permutation de $\{1, ..., 3\ell\}$. Il maintient un compteur $C_{\mathcal{R}}$, doit vérifier une signature digitale, et doit chiffrer un message de longueur $k + 3lg(3\ell)$ et deux messages de longueur $3\ell + k(3\ell + 1) + 3\ell lg(3\ell)$, où $3\ell lg(3\ell)$ est la longueur de la permutation. Il doit calculer le hachage d'un message de longueur $lg(c_{k+3lg(3\ell)})+$ $2lg(c_{3\ell+k(3\ell+1)+3\ell lg(3\ell)})$.

Au total, le lecteur choisit au hasard $2k(3\ell + 1) + 3\ell$ bits en plus de l'aléa nécessaire au choix d'une permutation de longueur 3ℓ et au chiffrement de deux messages de longueur $3\ell + k(3\ell + 1) + 3\ell lg(3\ell)$ et d'un message de longueur $k + 3lg(3\ell)$. Il envoie au serveur d'authentification un message de longueur $lg(c_{k+3lg(3\ell)}) + 2lg(c_{3\ell+k(3\ell+1)+3\ell lg(3\ell)})$.

• Le serveur $\mathcal{AS}$ doit déchiffrer un message de longueur $lg(c_{k+3lg(3\ell)})$, maintenir un compteur $C_{\mathcal{AS}}$, et calculer une signature digitale. Il doit calculer le hachage d'un message de longueur $lg(c_{k+3lg(3\ell)}) + 2lg(c_{3\ell+k(3\ell+1)+3\ell lg(3\ell)})$.

• Les bases de données doivent calculer (avec l'astuce de calcul vue en 2.4.3) 1 XOR d'au plus ℓ^3 flux morphométriques de longueur k et 3ℓ XORs d'au plus ℓ^2 flux morphométriques de longueur k.

4.3.4.4 Sécurité

4.3.4.4.1 Adversaires extérieurs

Intégrité de la réponse

Théorème 15 *Si le schéma de signature $\mathcal{DS}$ a des signatures inforgeables et la famille de fonctions de hachage H résiste aux collisions, le schéma d'authentification $\mathcal{PAM}^{semi-final}$ garantit l'intégrité de la réponse d'une ronde d'authentification contre les adversaires extérieurs.*

Soient n un paramètre de sécurité et $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4)$ un PPTA jouant contre $\mathcal{PAM}^{semi-final}$ dans le jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM}^{semi-final},\mathcal{B}}(n)$, et construisons $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ employant $\mathcal{B}$ pour attaquer $\mathfrak{H}$ ou $\mathcal{DS}$.

• *Phase de mise en place par $\mathcal{CH}$ des paramètres pour $\mathcal{A}$*

$\mathcal{CH}$ invoque $\mathcal{P}_s(1^n)$ pour obtenir I_s, puis $\mathcal{K}_s(I_s)$ pour calculer des clefs (pk_s, sk_s) pour $\mathcal{DS}$ puis $\mathcal{GH}(1^n)$ pour obtenir une fonction de hachage $\big(\mu, \langle H_\mu \rangle\big)$, et envoie $\big(pk_s, \langle H_\mu \rangle\big)$ à $\mathcal{A}_1$.

• *Phase de mise en place par $\mathcal{A}$ des paramètres pour $\mathcal{B}$*

$\mathcal{A}_1$ invoque $\mathcal{P}(1^n)$ pour obtenir I et invoque ensuite trois fois $\mathcal{K}(I)$ pour créer les paires de clefs $\big((pk_{\mathcal{AS}}, sk_{\mathcal{AS}}), (pk_0, sk_0), (pk_1, sk_1)\big)$ pour $\mathcal{CS}$.

• *Phase d'appels à l'oracle de signature par $\mathcal{A}$*

L'attaquant $\mathcal{A}_1$ répond aux requêtes de rondes de $\mathcal{B}_1$ à l'aide de requêtes de signatures faites à $\mathcal{CH}$:

Si

$$\Big(j_o^1 j_o^2 j_o^3, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big),$$

est la o-ième requête de $\mathcal{B}_1$, $\mathcal{A}_1$ simule tous les calculs d'une ronde d'authentification normale avec base de données $\big(k_o, No, \tau_o, (i, f_{oi})_i\big)$, flux de référence $f_{oj_o^1 j_o^2 j_o^3}$ et flux extrait $f_{oj_o^1 j_o^2 j_o^3} \oplus \kappa_o$. D'après les spécifications de $\mathcal{PAM}^{semi-final}$, $\mathcal{A}_1$ choisit deux masques M_{o0} et M_{o1} dans $\{0,1\}^{k_o(3\ell_o+1)}$ uniformément au hasard qu'il partitionne en $3\ell_o + 1$ blocs de longueur k_o :

$$M_{o0} = M_{o0}(1)||...||M_{o0}(3\ell_o + 1)$$

et

$$M_{o1} = M_{o1}(1)||...||M_{o1}(3\ell_o + 1),$$

choisit $\sigma_o \in \mathfrak{S}_{3\ell_o}$ uniformément au hasard, choisit S^1_{o0}, S^2_{o0} et S^3_{o0} dans l'ensemble des parties de $\{1, ..., \ell_o\}$ uniformément au hasard, calcule $S^1_{o1} := S_{o0} \oplus \{j^1_o\}$, $S^2_{o1} := S^2_{o0} \oplus \{j^2_o\}$ et $S^3_{o1} := S^3_{o0} \oplus \{j^3_o\}$, calcule $\sigma_o(j^1_o)$, $\sigma_o(\ell_o + j^2_o)$ et $\sigma_o(2\ell_o + j^3_o)$, calcule

$$M_o := \bigoplus_{b\in\{0,1\}} M_{ob}(j^1_o) \oplus M_{ob}(\ell_o + j^2_o) \oplus M_{ob}(2\ell_o + j^3_o) \oplus M_{ob}(3\ell_o + 1),$$

et

$$g_o := f_{oj^1_o j^2_o j^3_o} \oplus \kappa_o \oplus M_o,$$

invoque $\mathcal{E}_{pk_{\mathcal{AS}}}\big(g_o||\sigma_o(j^1_o)||\sigma_o(\ell_o + j^2_o)||\sigma_o(2\ell_o + j^3_o)\big)$ pour obtenir $c_{o\mathcal{AS}}$, invoque $\mathcal{E}_{pk_0}(S^1_{o0}||S^2_{o0}||S^3_{o0}||M_{o0}||\sigma_o)$ et $\mathcal{E}_{pk_1}(S^1_{o1}||S^2_{o1}||S^3_{o1}||M_{o1}||\sigma_o)$ pour obtenir c_{o0} et c_{o1} respectivement, incrémente le compteur du lecteur $C_{\mathcal{R}}$ (égal donc maintenant à o), pose

$$c_o = c_{o\mathcal{AS}}||c_{o0}||c_{o1}$$

et calcule $u_o := H_\mu(c_o||o)$. Ensuite, il détermine A_o en fonction de κ_o et τ_o, incrémente le compteur du serveur d'authentification $C_{\mathcal{AS}}$ (qui donc doit valoir o aussi) et envoie (A_o, u_o) comme requête de signature à $\mathcal{CH}$. Le challengeur $\mathcal{CH}$ invoque alors $Sign_{sk_s}(A_o, u_o)$ pour obtenir $\mathfrak{s}_o$ qui est retourné à $\mathcal{A}_1$. Ce dernier répond maintenant à $\mathcal{B}_1$ en lui donnant $\Big(o, c_o, \big((A_o, u_o), \mathfrak{s}_o\big)\Big)$.

Quand $\mathcal{B}_1$ arrête de faire des requêtes, $\mathcal{A}_1$ et $\mathcal{B}_1$ ont la liste suivante :

$$\mathfrak{Q} := \left(\Big(j^1_o j^2_o j^3_o, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big), \Big(o, c_o, \big(\big(j(\kappa_o, \tau_o), u_o\big), \mathfrak{s}_o\big)\Big)\right)_o$$

pour o dans $\{1, ..., q\}$ où q est majoré par un polynôme en n. La première partie d'un tel vecteur est l'information de la requête et la deuxième le résultat. Ce dernier contient comme réponse finale $A_o = j(\kappa_o, \tau_o)$ car les rondes sont valides.

$\mathcal{B}_1$ retourne la liste $\mathfrak{Q}$ ci-dessus et une information d'état e_1.

- *Requête de challenge de $\mathcal{B}$*

  $\mathcal{A}_1$ invoque $\mathcal{B}_2(\mathfrak{Q}, e_1)$ pour obtenir une requête de challenge

$$\Big(j^1 j^2 j^3, \big(k, N, \tau, (i, f_i)_i\big), \kappa, e_2\Big)$$

Cette requête peut déjà apparaître dans $\mathfrak{Q}$.

- *Première phase de réponse au challenge : $\mathcal{A}$ simule le premier message de $\mathcal{R}$*

  $\mathcal{A}_1$ choisit M_0 et M_1 uniformément au hasard dans $\{0,1\}^{k(3\ell+1)}$ et les partitionne en $3\ell + 1$ blocs de longueur k :

$$M_0 = M_0(1)||...||M_0(3\ell + 1)$$

et

$$M_1 = M_1(1)||...||M_1(3\ell+1),$$

, choisit $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard, choisit S_0^1, S_0^2 et S_0^3 uniformément au hasard parmi les sous-ensembles de $\{1,...,\ell\}$, calcule $S_1^1 := S_0^1 \oplus \{j^1\}$, $S_1^2 := S_0^2 \oplus \{j^2\}$ et $S_1^3 := S_0^3 \oplus \{j^3\}$, calcule calcule $S_1^1 := S_0 \oplus \{j^1\}$, $S_1^2 := S_0^2 \oplus \{j^2\}$ et $S_1^3 := S_0^3 \oplus \{j^3\}$, calcule $\sigma(j^1)$, $\sigma(\ell + j^2)$ et $\sigma(2\ell + j^3)$, calcule

$$M := \bigoplus_{b\in\{0,1\}} M_b(j^1) \oplus M_b(\ell + j^2) \oplus M_b(2\ell + j^3) \oplus M_b(3\ell + 1),$$

et

$$g_o := f_{j^1 j^2 j^3} \oplus \kappa \oplus M,$$

invoque $\mathcal{E}_{pk_{\mathcal{AS}}}\big(g||\sigma(j^1)||\sigma(\ell + j^2)||\sigma(2\ell + j^3)\big)$ pour obtenir $c_{\mathcal{AS}}$, invoque

$$\mathcal{E}_{pk_0}(S_0^1||S_0^2||S_0^3||M_0||\sigma) \text{ et } \mathcal{E}_{pk_1}(S_1^1||S_1^2||S_1^3||M_1||\sigma)$$

pour obtenir c_0 et c_1 respectivement, incrémente le compteur du lecteur $C_{\mathcal{R}}$ qui vaut donc $q+1$, pose

$$c = c_{\mathcal{AS}}||c_0||c_1$$

et calcule $u := H_\mu\big(c||(q+1)\big)$.

• *Deuxième phase de réponse au challenge : $\mathcal{B}$ manipule les messages de $\mathcal{R}$ vers $\mathcal{AS}$*

$\mathcal{A}_1$ invoque $\mathcal{B}_2(c_{\mathcal{AS}}, c_0, c_1, e_1)$ pour obtenir $\big(c'_{\mathcal{AS}}, c'_0, c'_1, , e_2\big)$ de sorte qu'il existe $g' \in \{0,1\}^k$, g'_1, g'_2 et g'_3 deux-à-deux distincts dans $\{1,...,3\ell\}$, L_0 et L_1 dans $\{0,1\}^{k(3\ell+1)+3\ell}$ et σ'_0 et σ'_1 dans $\mathfrak{S}_{3\ell}$ vérifiant $c'_{\mathcal{AS}} \in \mathcal{E}_{pk_{\mathcal{AS}}}(g'||g'_1||g'_3||g'_3)$, $c'_0 \in \mathcal{E}_{pk_0}(L_0||\sigma'_0)$ et $c'_0 \in \mathcal{E}_{pk_1}(L_1||\sigma'_1)$. Ces hypothèses sont nécessaires pour éviter que la ronde ne soit invalidée pour des raisons triviales de non-sens de calcul.

• *Troisième phase de réponse au challenge : $\mathcal{A}$ simule les calculs de l'interaction entre $\mathcal{AS}$ et les composantes $\mathcal{DB}_0$ et $\mathcal{DB}_1$*

$\mathcal{A}_1$ calcule g', g'_1, g'_2 et g'_3, ainsi que L_0, σ'_0, L_1 et σ'_1 avec les clefs secrètes. Il partitionne L_0 et L_1 en

$$S_0^{1\prime}||S_0^{2\prime}||S_0^{3\prime}||M'_0 \text{ et } S_1^{1\prime}||S_1^{2\prime}||S_1^{3\prime}||M'_1$$

respectivement, où M'_0 et M'_1 sont dans $\{0,1\}^{k(3\ell+1)}$ et les $S_b^{i\prime}$ sont dans $\{0,1\}^\ell$. Ensuite, $\mathcal{A}_1$ calcule

$$Z''_0 := X'_{100}(1)||...||X'_{100}(\ell)||X'_{010}(1)||...||X'_{010}(\ell)||X'_{001}(1)||...||X'_{001}(\ell)||X'_{000}$$

avec les $S_0^{i\prime}$ et

$$Z''_0 := X_{011}(1)'||...||X'_{011}(\ell)||X'_{101}(1)||...||X'_{101}(\ell)||X'_{110}(1)||...||X'_{110}(\ell)||X'_{111}$$

avec les $S_1^{i\prime}$. Puis, il calcule

$$Z'_0 := M'_0 \oplus Z''_0$$

et

$$Z'_1 := M'_1 \oplus Z''_1$$

et enfin il calcule $X'_0 := Z'_0$ auquel σ'_0 est appliquée aux 3ℓ premiers blocs de longueur k et $X'_1 := Z'_1$ auquel σ'_1 est appliquée aux 3ℓ premiers blocs de longueur k.

Les vecteurs binaires X'_0 et X'_1 sont tous deux dans $\{0,1\}^{k(3\ell+1)}$ et tout comme dans le protocole, pour tout $t \in \{1, ..., 3\ell+1\}$, on note $X'_0(t)$ (resp., $X'_1(t)$) le t-ième bloc en partant de la gauche de longueur k de X'_0 (resp., X'_1). Ainsi, on a

$$X'_0 = X'_0(1)||...||X'_0(3\ell+1)$$

et

$$X'_1 = X'_1(1)||...||X'_1(3\ell+1)$$

Pour terminer, $\mathcal{A}_1$ calcule

$$g' \oplus \bigoplus_{b\in\{0,1\}} X'_b(g'_1) \oplus X'_b(g'_2) \oplus X'_b(g'_3) \oplus X'_b(3\ell+1)$$

pour obtenir κ', dont il calcule le poids de Hamming et détermine sa réponse A' en fonction de ça et τ'. Enfin, il incrémente son compteur $C_{\mathcal{AS}}$ qui vaut donc $q+1$, pose $c' = c'_{\mathcal{AS}}||c'_0||c'_1$ et calcule $u' := H_\mu\big(c'||(q+1)\big)$. Il envoie (A', u') à $\mathcal{CH}$.

• *Quatrième phase de réponse au challenge : $\mathcal{A}$ obtient de $\mathcal{CH}$ une signature digitale pour simuler le message de $\mathcal{AS}$ à $\mathcal{R}$*

$\mathcal{CH}$ invoque $Sign_{sk_s}(A', u')$ pour obtenir $\mathfrak{s}'$ et envoie $\mathfrak{s}'$ à $\mathcal{A}_2$.

• *Réponse au challenge de $\mathcal{B}$: $\mathcal{B}$ manipule les messages de $\mathcal{AS}$ vers $\mathcal{R}$*

$\mathcal{A}_2$ invoque $\mathcal{B}_4\big((A', u'), \mathfrak{s}', e_3\big)$ pour obtenir $\big((A'', u''), \mathfrak{s}''\big)$ où $A'' \in \{0,1\}$.

• *Réponse au challenge de $\mathcal{A}$: $\mathcal{A}$ cherche dans ce qu'il a calculé une collision ou une signature forgée*

Si $A'' = j(\kappa, \tau) \oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'', u''), \mathfrak{s}''\big) = 1$, $\mathcal{A}_2$ pose $cm_{\mathcal{R}_2} = A''$. Puis :

- si $(A'', u'') = (A', u')$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c'||(q+1)\big)$ et
- si $(A'', u'') \neq (A', u')$:
  - si $(A'', u'') \notin \big\{(A_o, u_o)\big\}_o$, $\mathcal{A}$ retourne (A'', u'') et
  - s'il existe o tel que $(A'', u'') = (A_o, u_o)$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c_o||o\big)$.

Dans tous les autres cas, $\mathcal{A}_2$ retourne un message d'échec.

On voit d'abord que les trois conditions $A'' = j(\kappa,\tau) \oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'',u''),s''\big) = 1$ sont satisfaites si et seulement si $cm_{\mathcal{R}_2} = j(\kappa,\tau)\oplus 1$ à la fin du jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM}^{semi-final},\mathcal{B}}(n)$. Ensuite, que ces conditions soient satisfaites entraîne clairement soit $c' \neq c$ soit $(A'',u'') \neq (A',u')$ (sinon, $\mathcal{B}$ n'attaque pas le challenge et la réponse finale est $j(\kappa,\tau)$). Donc :

• si $(A'',u'') = (A',u')$, $c' \neq c$ et $\big(c'||(q+1), c||(q+1)\big)$ est une collision pour H_μ car $u'' = u$ et $u'' = u'$ donc

$$H_\mu\big(c'||(q+1)\big) = u' = u = H_\mu\big(c||(q+1)\big)$$

et

• si $(A'',u'') \neq (A',u')$, que $Verify_{pk_s}\big((A'',u''),\mathfrak{s}''\big) = 1$ implique que $\mathfrak{s}''$ est une signature valide sur le message (A'',u'') et il faut voir si (A'',u'') a déjà fait l'objet d'une requête de signature ou non.

Si $(A'',u'') \neq (A_o,u_o)$ pour tout o, (A'',u'') ne fait pas partie des signatures demandées à $\mathcal{CH}$ et donc c'est une signature forgée.

S'il existe o tel que $(A'',u'') = (A_o,u_o)$, on doit avoir $c||(q+1) \neq c_o||o$ car $o \leq q$ et on obtient une collision pour H_μ vu que

$$H_\mu(c_o||o) = u_o = u'' = u = H_\mu\big(c||(q+1)\big)$$

Il est clair que $\mathcal{A}$ joue contre $\mathfrak{H}$ et $\mathcal{DS}$ dans le jeu $\mathbf{Exp}^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}}$ d'où

$$Adv^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}} \geq Adv^{int-ext}_{\mathcal{PAM}^{semi-final},\mathcal{B}},$$

et le théorème est démontré. □

Confidentialité

Théorème 16 *Si $\mathcal{CS}$ est ind-cpa-sécurisé, $\mathcal{PAM}^{semi-final}$ garantit l'anonymat d'objet contre les adversaires extérieurs.*

Soient $\mathcal{A} := (\mathcal{A}_1,\mathcal{A}_2)$ un PPTA jouant contre $\mathcal{PAM}$ dans le jeu $\mathbf{Exp}^{oa-ext}_{\mathcal{PAM},\mathcal{A}}$, n le paramètre de sécurité et $\mathcal{CH}$ un challengeur. Le jeu se déroule comme suit.

Prenant 1^n en entrée, $\mathcal{CH}$ invoque une fois $\mathcal{P}(1^n)$ pour obtenir I et trois fois $\mathcal{K}(I)$ pour obtenir les trois clefs publiques $(pk_{\mathcal{AS}}, pk_0, pk_1)$ et les donne $\mathcal{A}_1$. Ce dernier calcule l'ensemble habituel $\Big(i_0, i_1, \big(k, N, \tau, (i,f_i)_i\big), \kappa\Big)$ qu'il donne à $\mathcal{CH}$. Puis, $\mathcal{CH}$ choisit

• $\beta \in \{0,1\}$ uniformément au hasard,
• M_0 et M_1 dans $\{0,1\}^{k(3\ell+1)}$ uniformément au hasard,
• S_0^1, S_0^2, S_0^3 dans $\{1,...,\ell\}$ uniformément au hasard et
• $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard,

calcule

• $M := \bigoplus_{b\in\{0,1\}} M_b(i^1_\beta) \oplus M_b(\ell + i^2_\beta) \oplus M_b(2\ell + i^3_\beta) \oplus M_b(3\ell+1)$,

- $g := M \oplus f_{i_\beta^1 i_\beta^2 i_\beta^3} \oplus \kappa$,
- $\sigma(i_\beta^1)$, $\sigma(\ell + i_\beta^2)$ et $\sigma(2\ell + i_\beta^3)$,
- $S_1^1 := S_0^1 \oplus \{i_\beta^1\}$, $S_1^1 := S_0^2 \oplus \{i_\beta^2\}$ et $S_1^3 := S_0^3 \oplus \{i_\beta^3\}$,
- $X_0 = X_0(1)||...||X_0(3\ell + 1)$ et
- $X_1 = X_1(1)||...||X_1(3\ell + 1)$,

invoque

- $\mathcal{E}_{pk_{\mathcal{AS}}}\big(g||\sigma(i_\beta^1)||\sigma(\ell + i_\beta^2)||\sigma(2\ell + i_\beta^3)\big)$ pour obtenir $c_{\mathcal{AS}}$,
- $\mathcal{E}_{pk_0}(S_0^1||S_0^2||S_0^3||M_0||\sigma)$ pour obtenir c_0 et
- $\mathcal{E}_{pk_1}(S_1^1||S_1^2||S_1^3||M_1||\sigma)$ pour obtenir c_1

et donne

$$\big((X_0, X_1), (c_{\mathcal{AS}}, c_0, c_1)\big)$$

à $\mathcal{A}_2$. Ce PPTA dispose donc de

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, (pk_{\mathcal{AS}}, pk_0, pk_1), (X_0, X_1), (c_{\mathcal{AS}}, c_0, c_1)\Big)$$

pour déterminer β. Cette interaction est aussi une instance du jeu $\mathbf{Exp}^{ind-cpia-api-mk}_{3,\mathcal{CS},\mathcal{M}_3,\mathcal{PI},\mathcal{A}}(n)$ donc toujours par les mêmes arguments, $Adv^{oa-ext}_{\mathcal{PAM},\mathcal{A}}$ sera prouvée négligeable dès que l'on aura prouvé que tout PPTA auquel on donne

$$\Big(i_0, i_1, \big(k, N, , \tau, (i, f_i)_i\big), \kappa, (X_0, X_1)\Big)$$

ne peut deviner β qu'avec une probabilité de $\frac{1}{2}$.

Maintenant, comme les M_0 et M_1 sont choisis uniformément au hasard, (X_0, X_1) est uniformément distribué. Le résultat en découle. □

4.3.4.4.2 Composantes malicieuses

Théorème 17 *Si $\mathcal{CS}$ est ind-cpa-sécurisé, $\mathcal{PAM}^{semi-final}$ garantit l'anonymat de l'objet contre des composantes malicieuses.*

Comme pour les autres protocoles, on partage cet énoncé en deux lemmes.

Lemme 9 *Si $\mathcal{CS}$ est ind-cpa-sécurisé et $\mathcal{CM} = \mathcal{AS}$, l'application*

$$Adv^{oa-comp}_{\mathcal{PAM}^{semi-final},\mathcal{CM}}$$

est négligeable.

Soient $\mathcal{CM} := \mathcal{AS}$ jouant contre $\mathcal{PAM}^{semi-final}$ dans le jeu $\mathbf{Exp}^{oa-ext}_{\mathcal{PAM}^{semi-final},\mathcal{A}}$, n le paramètre de sécurité et $\mathcal{CH}$ un challengeur. Le jeu se déroule comme suit.

Prenant 1^n en entrée, $\mathcal{CH}$ invoque une fois $\mathcal{P}(1^n)$ pour obtenir I et deux fois $\mathcal{K}(I)$ pour obtenir les deux clefs publiques (pk_0, pk_1) et les donne $\mathcal{CM}_1$. Ce

dernier calcule le quadruplet habituel $\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa\Big)$ qu'il donne à $\mathcal{CH}$. Puis, $\mathcal{CH}$ choisit

- $\beta \in \{0,1\}$ uniformément au hasard,
- M_0 et M_1 dans $\{0,1\}^{k(3\ell+1)}$ uniformément au hasard,
- S_0^1, S_0^2, S_0^3 dans $\{1, ..., \ell\}$ uniformément au hasard et
- $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard,

calcule

- $M := \bigoplus_{b\in\{0,1\}} M_b(i_\beta^1) \oplus M_b(\ell + i_\beta^2) \oplus M_b(2\ell + i_\beta^3) \oplus M_b(3\ell + 1)$,
- $g := M \oplus f_{i_\beta^1 i_\beta^2 i_\beta^3} \oplus \kappa$,
- $\sigma(i_\beta^1)$, $\sigma(\ell + i_\beta^2)$ et $\sigma(2\ell + i_\beta^3)$,
- $S_1^1 := S_0^1 \oplus \{i_\beta^1\}$, $S_1^1 := S_0^2 \oplus \{i_\beta^2\}$ et $S_1^3 := S_0^3 \oplus \{i_\beta^3\}$,
- $X_0 = X_0(1)||...||X_0(3\ell + 1)$ et
- $X_1 = X_1(1)||...||X_1(3\ell + 1)$,

invoque

- $\mathcal{E}_{pk_0}(S_0^1||S_0^2||S_0^3||M_0||\sigma)$ pour obtenir c_0 et
- $\mathcal{E}_{pk_1}(S_1^1||S_1^2||S_1^3||M_1||\sigma)$ pour obtenir c_1,

et donne

$$\big((g, X_0, X_1), (c_{\mathcal{AS}}, c_0, c_1)\big)$$

à $\mathcal{CM}_2$. Ce PPTA dispose donc de

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, (pk_0, pk_1), (g, X_0, X_1), (c_0, c_1)\Big)$$

pour déterminer β. Cette interaction est aussi une instance du jeu $\mathbf{Exp}^{ind-cpia-api-mk}_{2,\mathcal{CS},\mathcal{M}_2,\mathcal{PI},\mathcal{CM}}(n)$ donc toujours par les mêmes arguments, $Adv^{oa-ext}_{\mathcal{PAM}^{semi-final},\mathcal{CM}}$ sera prouvée négligeable dès que l'on aura prouvé que tout PPTA auquel on donne

$$\Big(i_0, i_1, (k, N, \tau, (i, f_i)_i), \kappa, (g, X_0, X_1)\Big)$$

ne peut deviner β qu'avec une probabilité de $\frac{1}{2}$.

Comme les M_0 et M_1 sont choisis uniformément au hasard, l'octuplet

$$\Big(\big(X_b(\sigma(i_\beta^1), X_b(\sigma(\ell + i_\beta^2)), X_b(\sigma(2\ell + i_\beta^3)), X_b(\sigma(3\ell))\big)_{b\in\{0,1\}}\Big)$$

obtenu à partir de $\sigma(i_\beta^1)$, $\sigma(\ell+i_\beta^2)$ et $\sigma(2\ell+i_\beta^3)$ est uniformément distribué parmi ceux de somme κ, et les morceaux "restants" de X_0 et X_1 en sont indépendants et uniformément distribués dans $\{0,1\}^{k(3\ell-3)}$. Le résultat en découle. □

Lemme 10 *Si $\mathcal{CM} \in \{\mathcal{DB}_0, \mathcal{DB}_1\}$, l'application*

$$Adv^{oa-comp}_{\mathcal{PAM}^{semi-final},\mathcal{CM}}$$

est nulle donc négligeable.

Soit $\mathcal{CM}$ l'une des deux bases de données, qu'on peut sans perte de généralité supposer être $\mathcal{CM} = \mathcal{BD}_0$.

On peut simuler alors $\mathbf{Exp}^{oa-comp}_{\mathcal{PAM}^{semi-final},\mathcal{CM}}(n)$ pour un $n \in \mathbb{N}$: $\mathcal{CM}$ prend en entrée 1^n et calcule $\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa\Big)$ qu'il donne au challengeur $\mathcal{CH}$. Ce dernier choisit $\beta \in \{0,1\}$ uniformément au hasard, choisit S_0^1, S_0^2 et S_0^3 dans $\{1, ..., \ell\}$, uniformément au hasard, choisit $M_0 \in \{0,1\}^k$ uniformément au hasard, choisit $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard et envoie $S_0^1||S_0^2||S_0^3||M_0||\sigma$ à $\mathcal{CM}$. Le PPTA $\mathcal{CM}$ possède donc

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, S_0^1||S_0^2||S_0^3||M_0||\sigma\Big)$$

pour déterminer β. Vu la distribution uniforme de S_0^1, S_0^2 et S_0^3, la probabilité que $\mathcal{CM}$ devine β est $\frac{1}{2}$, i.e. $Adv^{oa-comp}_{\mathcal{PAM}^{semi-final},\mathcal{CM}}$ est nulle. □

4.3.4.5 Le protocole final : la ronde d'authentification

Enfin, nous décrivons le dernier protocole imaginé, appelé $\mathcal{PAM}^{final}$. Il ne diffère du protocole $\mathcal{PAM}^{semi-final}$ ci-dessus qu'au niveau de la ronde d'authentification. Toute l'infrastructure est identique donc nous donnons tout de suite la description de la ronde.

Pour authentifier l'objet $\mathcal{O}'$ prétendant être $\mathcal{O}$ enregistré au sein du système à l'indice $i_{\mathcal{O}}$, la procédure se déroule comme suit :

1) $\mathcal{R}$ lit $i_{\mathcal{O}}$ et extrait f' ;

2) $\mathcal{R}$ entame le PIR (deux bases de données en simulant huit) :

a) $\mathcal{R}$ obtient (i^1, i^2, i^3) de $i_{\mathcal{O}}$;

b) $\mathcal{R}$ choisit S_0^1, S_0^2, et S_0^3 dans $\{1, ..., \ell\}$ uniformément au hasard ;

c) $\mathcal{R}$ calcule $S_1^1 := S_0^1 \oplus \{i^1\}$, $S_1^2 := S_0^2 \oplus \{i^2\}$, et $S_1^3 := S_0^3 \oplus \{i^3\}$;

d) $\mathcal{R}$ choisit huit masques $M_{s_1 s_2 s_3} \in \{0,1\}^k$ uniformément au hasard ;

e) $\mathcal{R}$ choisit $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard ;

f) $\mathcal{R}$ calcule

$$c_0 := \mathcal{E}_{pk_0}(S_0^1||S_0^2||S_0^3||M_{000}||M_{100}||M_{010}||M_{001}||\sigma),$$

$$c_1 := \mathcal{E}_{pk_1}(S_1^1||S_1^2||S_1^3||M_{111}||M_{011}||M_{101}||M_{110}||\sigma),$$

et

$$c_{\mathcal{AS}} := \mathcal{E}_{pk_{\mathcal{AS}}}\big(f' \oplus \bigoplus_{s_1 s_2 s_3} M_{s_1 s_2 s_2}||\sigma(i^1)||\sigma(\ell + i^2)||\sigma(2\ell + i^3)\big);$$

$\mathcal{R}$ envoie $(c_0, c_1, c_{\mathcal{AS}})$ à $\mathcal{AS}$;

3) $\mathcal{AS}$ transmet c_0 à $\mathcal{DB}_0$ et c_1 à $\mathcal{DB}_1$;

4) $\mathcal{DB}_0$ (resp., $\mathcal{DB}_1$) calcule

$$S_0^1||S_0^2||S_0^3||M_{000}||M_{100}||M_{010}||M_{001}||\sigma$$

(resp.,

$$S_1^1||S_1^2||S_1^3||M_{111}||M_{011}||M_{101}||M_{110}||\sigma),$$

puis il calcule :

a)

$$X_0'' := X_{100}(1)||...||X_{100}(\ell)||X_{010}(1)||...||X_{010}(\ell)||X_{001}(1)||...||X_{001}(\ell)||X_{000}$$

(resp.,

$$X_1'' := X_{011}(1)||...||X_{011}(\ell)||X_{101}(1)||...||X_{101}(\ell)||X_{110}(1)||...||X_{110}(\ell)||X_{111});$$

b)

$$X_0' := M_{100}||...||M_{100}||M_{010}||...||M_{010}||M_{001}||...||M_{001}||M_{000} \oplus X_0''$$

(resp.,

$$X_1' := M_{011}||...||M_{011}||M_{101}||...||M_{101}||M_{110}||...||M_{110}||M_{111} \oplus X_1'');$$

c) $X_0 := X_0'$ auquel σ est appliquée aux 3ℓ premiers blocs de longueur k (de même pour X_1 avec X_1').

Les vecteurs X_0 et X_1 sont tous deux dans $\{0,1\}^{k(3\ell+1)}$. Pour tout $t \in \{1,...,3\ell+1\}$, on note $X_0(t)$ (resp., $X_1(t)$) le t-ième bloc en partant de la gauche de longueur k de X_0 (resp., X_1). Ainsi, on a

$$X_0 = X_0(1)||...||X_0(3\ell+1)$$

et

$$X_1 = X_1(1)||...||X_1(3\ell+1)$$

$\mathcal{DB}_0$ calcule $d_0 := \mathcal{E}_{pk_{\mathcal{AS}}}(X_0)$ (resp., $\mathcal{DB}_1$ calcule $d_1 := \mathcal{E}_{pk_{\mathcal{AS}}}(X_1)$) et envoie d_0 (resp., d_1) à $\mathcal{AS}$;

5) $\mathcal{AS}$ calcule $g := f' \oplus \bigoplus_{s_1s_2s_3} M_{s_1s_2s_3}$, $\sigma(i^1)$,$\sigma(\ell+i^2)$ et $\sigma(2\ell+i^3)$, obtient X_0 et X_1 de d_0 et d_1 et calcule

$$g \oplus \bigoplus_{b\in\{0,1\}} X_b\big(\sigma(i^1)\big) \oplus X_b\big(\sigma(\ell+i^2)\big) \oplus X_b\big(\sigma(2\ell+i^3)\big) \oplus X_b(3\ell+1)$$

qui n'est autre que $f \oplus f'$. Enfin, $\mathcal{AS}$ calcule le poids de Hamming de $f \oplus f'$ pour obtenir une réponse A, invoque $Sign_{sk_s}(A,u)$ pour obtenir une signature digitale $\mathfrak{s}$ sur (A,u) et envoie $\big((A,u),\mathfrak{s}\big)$ à $\mathcal{R}$.

6) $\mathcal{R}$ regarde si le u qu'il a gardé en mémoire est égal à celui envoyé par $\mathcal{AS}$. Il vérifie ensuite la signature digitale $\mathfrak{s}$ sur $\big((A,u),\mathfrak{s}\big)$. Si l'une de ces vérifications échoue, $\mathcal{R}$ retourne $\perp$ et invalide la ronde. Sinon, il accepte A.

4.3.4.6 Coût en calculs

Regardons le travail des composantes individuelles.

• Le lecteur $\mathcal{R}$ doit choisir de manière pseudo-aléatoire huit vecteurs binaires de longueur k, trois vecteurs binaires de longueur ℓ, et une permutation de $\{1,...,3\ell\}$. Il maintient un compteur $C_{\mathcal{R}}$, doit vérifier une signature digitale, et doit chiffrer un message de longueur $k + 3lg(3\ell)$ et deux messages de longueur $3\ell + 4k + 3\ell lg(3\ell)$, où $3\ell lg(3\ell)$ est la longueur de la permutation. Il doit calculer le hachage d'un message de longueur $lg(c_{k+3lg(3\ell)}) + 2lg(c_{3\ell+4k+3\ell lg(3\ell)})$.

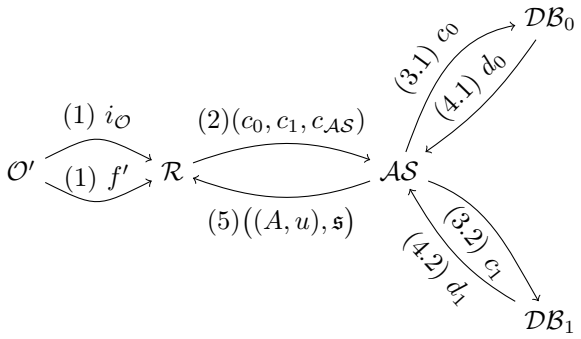


FIGURE 4.3 – Le protocole final

Au total, le lecteur choisit au hasard $8k + 3\ell$ bits en plus de l'aléa nécessaire au choix d'une permutation de longueur 3ℓ et au chiffrement de deux messages de longueur $3\ell + 4k + 3\ell lg(3\ell)$ et d'un message de longueur $k + 3lg(3\ell)$. Il envoie au serveur d'authentification un message de longueur $lg(c_{k+3lg(3\ell)}) + 2lg(c_{3\ell+4k+3\ell lg(3\ell)})$.

• Le serveur $\mathcal{AS}$ doit déchiffrer un message de longueur $lg(c_{k+3lg(3\ell)})$ et deux messages de longueur $lg(c_{k(3\ell+1)})$, maintenir un compteur $C_{\mathcal{AS}}$, et calculer une signature digitale. Il doit calculer le hachage d'un message de longueur $lg(c_{k+3lg(3\ell)}) + 2lg(c_{3\ell+4k+3\ell lg(3\ell)})$.

• Les bases de données doivent déchiffrer un message de longueur $lg(c_{3_ell+4k+3\ell lg(3\ell)})$ calculer 1 XOR d'au plus ℓ^3 flux morphométriques de longueur k et 3ℓ XORs d'au plus ℓ^2 flux morphométriques de longueur k. Elles doivent aussi chiffrer un message de longueur $k(3\ell + 1)$.

4.3.4.7 Sécurité

4.3.4.7.1 Adversaires extérieurs

Intégrité de la réponse finale

Théorème 18 *Si le schéma de signature $\mathcal{DS}$ a des signatures inforgeables et la famille de fonctions de hachage $\mathfrak{H}$ résiste aux collisions, le schéma d'authentification $\mathcal{PAM}^{final}$ garantit l'intégrité de la réponse d'une ronde d'authentification contre les adversaires extérieurs.*

Soit $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4)$ un PPTA fonctionnant en quatre étapes et jouant contre $\mathcal{PAM}^{final}$ dans le jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM}^{final},\mathcal{B}}(n)$ pour n un paramètre de sécurité et construisons $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ utilisant $\mathcal{B}$ comme sous-programme pour attaquer $\mathfrak{H}$ ou $\mathcal{DS}$.

- *Phase de mise en place par $\mathcal{CH}$ des paramètres pour $\mathcal{A}$*

Le challengeur $\mathcal{CH}$ invoque $\mathcal{P}_s(1^n)$ pour avoir I_s, $\mathcal{K}_s(I_s)$ pour avoir les clefs (pk_s, sk_s) de $\mathcal{DS}$ et $\mathcal{GH}(1^n)$ pour obtenir $\big(\mu, \langle H_\mu\rangle\big)$, et envoie $\big(pk_s, \langle H_\mu\rangle\big)$ à $\mathcal{A}_1$.

- *Phase de mise en place par $\mathcal{A}$ des paramètres pour $\mathcal{B}$*

$\mathcal{A}_1$ invoque $\mathcal{P}(1^n)$ pour avoir I et trois fois $\mathcal{K}(I)$ pour avoir les clefs de chiffrement $\big((pk_{\mathcal{AS}}, sk_{\mathcal{AS}}), (pk_0, sk_0), (pk_1, sk_1)\big)$ pour $\mathcal{CS}$.

- *Phase d'appels à l'oracle de signature par $\mathcal{A}$*

$\mathcal{A}_1$ répond aux requêtes de rondes d'authentifications de $\mathcal{B}_1$ avec ses propres requêtes de signature à $\mathcal{CH}$. Il invoque $\mathcal{B}_1\big(pk_s, pk_{\mathcal{AS}}, pk_0, \langle H_\mu\rangle\big)$ et gère les demandes de ce dernier comme suit.

Quand $\mathcal{B}_1$ spécifie une o-ième requête de la forme

$$\Big(j_o^1 j_o^2 j_o^3, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big),$$

$\mathcal{A}_1$ simule tous les calculs d'une ronde d'authentification valide avec comme entrées la base de données $\big(k_o, N_o, \tau_o, (i, f_{oi})_i\big)$, le flux de référence $f_{oj_o^1 j_o^2 j_o^3}$ et le flux extrait $f_{oj_o^1 j_o^2 j_o^3} \oplus \kappa_o$. Vu les spécifications, $\mathcal{A}_1$ choisit huit masques $M_{os_1s_2s_3}$ dans $\{0,1\}^k$ uniformément au hasard, choisit $\sigma_o \in \mathfrak{S}_{3\ell_o}$ uniformément au hasard, choisit S_{o0}^1, S_{o0}^2 et S_{o0}^3 dans l'ensemble des parties de $\{1, ..., \ell_o\}$ uniformément au hasard, calcule $S_{o1}^1 := S_{o0} \oplus \{j_o^1\}$, $S_{o1}^2 := S_{o0}^2 \oplus \{j_o^2\}$ et $S_{o1}^3 := S_{o0}^3 \oplus \{j_o^3\}$, calcule $\sigma_o(j_o^1)$, $\sigma_o(\ell_o + j_o^2)$ et $\sigma_o(2\ell_o + j_o^3)$, calcule

$$g_o := f_{oj_o^1 j_o^2 j_o^3} \oplus \kappa_o \oplus \bigoplus_{s_1 s_2 s_3} M_{os_1 s_2 s_3},$$

invoque $\mathcal{E}_{pk_{\mathcal{AS}}}\big(g_o||\sigma_o(j_o^1)||\sigma_o(\ell_o + j_o^2)||\sigma_o(2\ell_o + j_o^3)\big)$ pour obtenir $c_{o\mathcal{AS}}$, invoque

$$\mathcal{E}_{pk_0}(S_{o0}^1||S_{o0}^2||S_{o0}^3||M_{o000}||M_{o100}||M_{o010}||M_{o001}||\sigma_o)$$

et

$$\mathcal{E}_{pk_1}(S_{o1}^1||S_{o1}^2||S_{o1}^3||M_{o111}||M_{o011}||M_{o101}||M_{o110}||\sigma_o)$$

pour obtenir c_{o0} et c_{o1} respectivement, incrémente le compteur du lecteur $C_{\mathcal{R}}$ qui est donc maintenant égal à o, pose

$$c_o = c_{o\mathcal{AS}}||c_{o0}||c_{o1}$$

et calcule $u_o := H_\mu(c_o||o)$. Il calcule A_o à l'aide de κ_o et τ_o, incrémente le compteur $C_{\mathcal{AS}}$ du serveur (qui vaut donc maintenant o aussi) et envoie (A_o, u_o) comme requête à $\mathcal{CH}$. Le challengeur $\mathcal{CH}$ invoque alors $Sign_{sk_s}(A_o, u_o)$ pour avoir $\mathfrak{s}_o$ qui est retourné à $\mathcal{A}_1$. Ce dernier répond maintenant à $\mathcal{B}_1$ en lui donnant $\Big(o, c_o, \big((A_o, u_o), \mathfrak{s}_o\big)\Big)$.

Quand cette phase s'arrête à la demande de $\mathcal{B}_1$, $\mathcal{A}_1$ et $\mathcal{B}_1$ possèdent :

$$\mathfrak{Q} := \left(\Big(j_o^1 j_o^2 j_o^3, \big(k_o, N_o, \tau_o, (i, f_{oi})_i\big), \kappa_o\Big), \left(o, c_o, \Big(\big(j(\kappa_o, \tau_o), u_o\big), \mathfrak{s}_o\Big)\right)\right)_o$$

pour o parcourant $\{1, ..., q\}$ où q est majoré par un polynôme en n. La réponse finale de chaque requête est $A_o = j(\kappa_o, \tau_o)$ car les rondes sont valides.

Enfin, $\mathcal{B}_1$ retourne la liste $\mathfrak{Q}$ ci-dessus et une information d'état e_1.

• *Requête de challenge de $\mathcal{B}$*

$\mathcal{A}_1$ invoque $\mathcal{B}_2(\mathfrak{Q}, e_1)$ pour avoir une requête de challenge

$$\Big(j^1 j^2 j^3, \big(k, \ell, \tau, (i, f_i)_i\big), \kappa, e_2\Big)$$

• *Première phase de réponse au challenge : $\mathcal{A}$ simule le premier message de $\mathcal{R}$*

$\mathcal{A}_1$ choisit maintenant les masques $M_{s_1 s_2 s_3}$ uniformément au hasard dans $\{0,1\}^k$, choisit $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard, choisit S_0^1, S_0^2 et S_0^3 uniformément au hasard parmi les sous-ensembles de $\{1, ..., \ell\}$, calcule $S_1^1 := S_0^1 \oplus \{j^1\}$, $S_1^2 := S_0^2 \oplus \{j^2\}$ et $S_1^3 := S_0^3 \oplus \{j^3\}$, calcule $\sigma(j^1)$, $\sigma(\ell + j^2)$ et $\sigma(2\ell + j^3)$, calcule

$$g_o := f_{j^1 j^2 j^3} \oplus \kappa \oplus \bigoplus_{s_1 s_2 s_3} M_{s_1 s_2 s_3},$$

invoque $\mathcal{E}_{pk_{\mathcal{AS}}}\big(g||\sigma(j^1)||\sigma(\ell + j^2)||\sigma(2\ell + j^3)\big)$ pour obtenir $c_{\mathcal{AS}}$, invoque

$$\mathcal{E}_{pk_0}(S_0^1||S_0^2||S_0^3||M_{000}||M_{100}||M_{010}||M_{001}||\sigma)$$

et

$$\mathcal{E}_{pk_1}(S_1^1||S_1^2||S_1^3||M_{111}||M_{011}||M_{101}||M_{110}||\sigma)$$

pour obtenir c_0 et c_1 respectivement, incrémente le compteur du lecteur $C_{\mathcal{R}}$ qui vaut donc $q + 1$, pose

$$c = c_{\mathcal{AS}}||c_0||c_1$$

et calcule $u := H_\mu\big(c||(q+1)\big)$.

• *Deuxième phase de réponse au challenge : $\mathcal{B}$ manipule les messages de $\mathcal{R}$ vers $\mathcal{AS}$*

$\mathcal{A}_1$ invoque $\mathcal{B}_2(c_{\mathcal{AS}}, c_0, c_1, e_1)$ pour obtenir $\big(c'_{\mathcal{AS}}, c'_0, c'_1, e_2\big)$ de sorte qu'il existe $g' \in \{0,1\}^k$, g'_1, g'_2 et g'_3 deux-à-deux distincts dans $\{1, ..., 3\ell\}$, L_0 et L_1 dans $\{0,1\}^{3\ell+4k}$ et σ'_0 et σ'_1 dans $\mathfrak{S}_{3\ell}$ vérifiant $c'_{\mathcal{AS}} \in \mathcal{E}_{pk_{\mathcal{AS}}}(g'||g'_1||g'_3||g'_3)$, $c'_0 \in \mathcal{E}_{pk_0}(L_0||\sigma'_0)$ et $c'_1 \in \mathcal{E}_{pk_1}(L_1||\sigma'_1)$. Comme d'habitude, ces hypothèses servent à ne pas fausser prématurément la ronde pour des raisons d'impossibilité de calcul.

• *Troisième phase de réponse au challenge : $\mathcal{A}$ simule les calculs de l'interaction entre $\mathcal{AS}$ et les composantes $\mathcal{DB}_0$ et $\mathcal{DB}_1$*

$\mathcal{A}_1$ calcule g', g'_1, g'_2 et g'_3, ainsi que L_0, σ'_0, L_1 et σ'_1 avec les clefs secrètes. Il partitionne L_0 et L_1 en

$$S_0^{1\prime}||S_0^{2\prime}||S_0^{3\prime}||M'_{000}||M'_{100}||M'_{010}||M'_{001}$$

et

$$S_1^{1\prime}||S_1^{2\prime}||S_1^{3\prime}||M'_{111}||M'_{011}||M'_{101}||M'_{110}$$

respectivement, où les $M'_{s_1s_2s_3}$ sont dans $\{0,1\}^k$ et les $S_b^{i\prime}$ sont dans $\{0,1\}^\ell$. Ensuite, $\mathcal{A}_1$ calcule

$$Z''_0 := X_{100}(1)'||...||X'_{100}(\ell)||X'_{010}(1)||...||X'_{010}(\ell)||X'_{001}(1)||...||X'_{001}(\ell)||X'_{000}$$

avec les $S_0^{i\prime}$ et

$$Z''_0 := X_{011}(1)'||...||X'_{011}(\ell)||X'_{101}(1)||...||X'_{101}(\ell)||X'_{110}(1)||...||X'_{110}(\ell)||X'_{111}$$

avec les $S_1^{i\prime}$. Puis, il calcule

$$Z'_0 := M'_{100}||...||M'_{100}||M'_{010}||...||M'_{010}||M'_{001}||...||M'_{001}||M'_{000} \oplus Z''_0$$

et

$$Z'_1 := M'_{011}||...||M'_{011}||M'_{101}||...||M'_{101}||M'_{110}||...||M'_{110}||M'_{111} \oplus Z''_1$$

et enfin il calcule $X'_0 := Z'_0$ auquel σ'_0 est appliquée aux 3ℓ premiers blocs de longueur k et $X'_1 := Z'_1$ auquel σ'_1 est appliquée aux 3ℓ premiers blocs de longueur k.

Les vecteurs binaires X'_0 et X'_1 sont tous deux dans $\{0,1\}^{k(3\ell+1)}$ et tout comme dans le protocole, pour tout $t \in \{1,...,3\ell+1\}$, on note $X'_0(t)$ (resp., $X'_1(t)$) le t-ième bloc en partant de la gauche de longueur k de X'_0 (resp., X'_1). Ainsi, on a

$$X'_0 = X'_0(1)||...||X'_0(3\ell+1)$$

et

$$X'_1 = X'_1(1)||...||X'_1(3\ell+1)$$

Pour terminer, $\mathcal{A}_1$ calcule

$$g' \oplus \bigoplus_{b\in\{0,1\}} X'_b(g'_1) \oplus X'_b(g'_2) \oplus X'_b(g'_3) \oplus X'_b(3\ell+1)$$

pour obtenir κ', dont il calcule le poids de Hamming et détermine sa réponse A' en fonction de ça et τ'. Enfin, il incrémente son compteur $C_{\mathcal{AS}}$ qui vaut donc $q+1$, pose $c' = c'_{\mathcal{AS}}||c'_0||c'_1$ et calcule $u' := H_\mu\big(c'||(q+1)\big)$. Il envoie (A',u') à $\mathcal{CH}$.

• *Quatrième phase de réponse au challenge : $\mathcal{A}$ obtient de $\mathcal{CH}$ une signature digitale pour simuler le message de $\mathcal{AS}$ à $\mathcal{R}$*

$\mathcal{CH}$ invoque $Sign_{sk_s}(A',u')$ pour obtenir $\mathfrak{s}'$ et envoie $\mathfrak{s}'$ à $\mathcal{A}_2$.

• *Réponse au challenge de $\mathcal{B}$: $\mathcal{B}$ manipule les messages de $\mathcal{AS}$ vers $\mathcal{R}$*

$\mathcal{A}_2$ invoque $\mathcal{B}_4\big((A',u'),\mathfrak{s}',e_3\big)$ pour obtenir $\big((A'',u''),\mathfrak{s}''\big)$ où $A'' \in \{0,1\}$.

• *Réponse au challenge de $\mathcal{A}$: $\mathcal{A}$ cherche dans ce qu'il a calculé une collision ou une signature forgée*

Si $A'' = j(\kappa,\tau)\oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'',u''),\mathfrak{s}''\big) = 1$, $\mathcal{A}_2$ pose $cm_{\mathcal{R}_2} = A''$. Puis :

- si $(A'',u'') = (A',u')$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c'||(q+1)\big)$ et
- si $(A'',u'') \neq (A',u')$:
  - si $(A'',u'') \notin \big\{(A_o,u_o)\big\}_o$, $\mathcal{A}$ retourne (A'',u'') et
  - s'il existe o tel que $(A'',u'') = (A_o,u_o)$, $\mathcal{A}_2$ retourne $\big(c||(q+1), c_o||o\big)$.

Dans tous les autres cas, $\mathcal{A}_2$ retourne un message d'échec.

Les trois conditions $A'' = j(\kappa,\tau)\oplus 1$, $u'' = u$ et $Verify_{pk_s}\big((A'',u''),s''\big) = 1$ sont satisfaites simultanément si et seulement si $cm_{\mathcal{R}_2} = j(\kappa,\tau)\oplus 1$ à la fin du jeu $\mathbf{Exp}^{int-ext}_{\mathcal{PAM},\mathcal{B}}(n)$. Puis, si ces conditions sont satisfaites, on a soit $c' \neq c$ soit $(A'',u'') \neq (A',u')$ faute de quoi $\mathcal{B}$ n'attaque pas le challenge et la réponse finale de l'authentification est $j(\kappa,\tau)$. Il en résulte que :

• si $(A'',u'') = (A',u')$, $c' \neq c$ et $\big(c'||(q+1), c||(q+1)\big)$ est une collision pour H_μ car $u'' = u$ et $u'' = u'$ donc

$$H_\mu\big(c'||(q+1)\big) = u' = u = H_\mu\big(c||(q+1)\big)$$

et

• si $(A'',u'') \neq (A',u')$, que $Verify_{pk_s}\big((A'',u''),\mathfrak{s}''\big) = 1$ implique que $\mathfrak{s}''$ est une signature valide sur le message (A'',u'') et il reste à voir si (A'',u'') a déjà fait l'objet d'une requête de signature ou non.

Si $(A'',u'') \neq (A_o,u_o)$ pour tout o, (A'',u'') ne fait pas partie des signatures demandées à $\mathcal{CH}$ et donc c'est une signature forgée.

S'il existe o tel que $(A'',u'') = (A_o,u_o)$, on doit avoir $c||(q+1) \neq c_o||o$ car $o \leq q$ et on obtient une collision pour H_μ vu que

$$H_\mu(c_o||o) = u_o = u'' = u = H_\mu\big(c||(q+1)\big)$$

Comme $\mathcal{A}$ joue contre $\mathfrak{H}$ et $\mathcal{DS}$ dans le jeu $\mathbf{Exp}^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}}$, on peut conclure :

$$Adv^{coll/for}_{\mathfrak{H},\mathcal{DS},\mathcal{A}} \geq Adv^{int-ext}_{\mathcal{PAM},\mathcal{B}}$$

ce qui achève la preuve. □

Confidentialité

Théorème 19 *Si $\mathcal{CS}$ est ind-cpa-sécurisé, $\mathcal{PAM}^{final}$ garantit l'anonymat d'objet contre les adversaires extérieurs.*

Soient $\mathcal{A} := (\mathcal{A}_1,\mathcal{A}_2)$ un PPTA jouant contre $\mathcal{PAM}^{final}$ dans le jeu $\mathbf{Exp}^{oa-ext}_{\mathcal{PAM}^{final},\mathcal{A}}$, n le paramètre de sécurité et $\mathcal{CH}$ un challengeur. Le jeu se déroule comme suit.

Prenant 1^n en entrée, $\mathcal{CH}$ invoque une fois $\mathcal{P}(1^n)$ pour obtenir I et trois fois $\mathcal{K}(I)$ pour obtenir les trois clefs publiques $(pk_{\mathcal{AS}}, pk_0, pk_1)$ et les donne $\mathcal{A}_1$. Ce

dernier calcule l'ensemble habituel $\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa\Big)$ qu'il donne à $\mathcal{CH}$. Puis, $\mathcal{CH}$ choisit

- $\beta \in \{0,1\}$ uniformément au hasard,
- huit masques $M_{s_1 s_2 s_3}$ dans $\{0,1\}^k$ uniformément au hasard,
- S_0^1, S_0^2, S_0^3 dans $\{1, ..., \ell\}$ uniformément au hasard et
- $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard,

calcule

- $g := f_{i_\beta^1 i_\beta^2 i_\beta^3} \oplus \kappa \oplus \bigoplus_{s_1 s_2 s_3} M_{s_1 s_2 s_3}$,
- $\sigma(i_\beta^1)$, $\sigma(\ell + i_\beta^2)$ et $\sigma(2\ell + i_\beta^3)$,
- $S_1^1 := S_0^1 \oplus \{i_\beta^1\}$, $S_1^1 := S_0^2 \oplus \{i_\beta^2\}$ et $S_1^3 := S_0^3 \oplus \{i_\beta^3\}$,
- $X_0 = X_0(1)||...||X_0(3\ell + 1)$ et
- $X_1 = X_1(1)||...||X_1(3\ell + 1)$,

invoque

- $\mathcal{E}_{pk_{\mathcal{AS}}}\big(g||\sigma(i_\beta^1)||\sigma(\ell + i_\beta^2)||\sigma(2\ell + i_\beta^3)\big)$ pour obtenir $c_{\mathcal{AS}}$,
- $\mathcal{E}_{pk_{\mathcal{AS}}}(X_0)$ pour obtenir d_0,
- $\mathcal{E}_{pk_{\mathcal{AS}}}(X_1)$ pour obtenir d_1,
- $\mathcal{E}_{pk_0}(S_0^1||S_0^2||S_0^3||M_0||\sigma)$ pour obtenir c_0 et
- $\mathcal{E}_{pk_1}(S_1^1||S_1^2||S_1^3||M_1||\sigma)$ pour obtenir c_1

et donne

$$(c_{\mathcal{AS}}, d_0, d_1, c_0, c_1)$$

à $\mathcal{A}_2$. Ce PPTA dispose donc de

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, (pk_{\mathcal{AS}}, pk_0, pk_1), (c_{\mathcal{AS}} d_0, d_1, c_0, c_1)\Big)$$

pour déterminer β. Cette interaction est aussi une instance du jeu $\mathbf{Exp}_{3,\mathcal{CS},\mathcal{M}_3,\mathcal{PI},\mathcal{A}}^{ind-cpia-api-mk}(n)$ donc toujours par les mêmes arguments, $Adv_{\mathcal{PAM}^{final},\mathcal{A}}^{oa-ext}$ est négligeable car tout PPTA auquel on donne

$$\Big(i_0, i_1, (k, N, \tau, (i, f_i)_i), \kappa\Big)$$

ne peut deviner β qu'avec une probabilité de $\frac{1}{2}$. □

4.3.4.7.2 Composantes malicieuses

Une base de données malicieuse

Lemme 11 *Si $\mathcal{CM} \in \{\mathcal{DB}_0, \mathcal{DB}_1\}$, l'application*

$$Adv_{\mathcal{PAM}^{final},\mathcal{CM}}^{oa-comp}$$

est nulle donc négligeable.

Soit $\mathcal{CM}$ l'une des deux bases de données, qu'on peut sans perte de généralité être $\mathcal{CM} = \mathcal{DB}_0$.

On peut simuler alors $\mathbf{Exp}^{oa-comp}_{\mathcal{PAM}^{final},\mathcal{CM}}(n)$ pour un $n \in \mathbb{N}$: $\mathcal{CM}$ prend en entrée 1^n et calcule $\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa\Big)$ qu'il donne au challengeur $\mathcal{CH}$. Ce dernier choisit $\beta \in \{0,1\}$ uniformément au hasard, choisit S_0^1, S_0^2 et S_0^3 dans $\{1, ..., \ell\}$, uniformément au hasard, choisit quatre masques $M_{s_1s_2s_3} \in \{0,1\}^k$ uniformément au hasard pour $s_1s_2s_3 \in \{000, 001, 010, 100\}$, choisit $\sigma \in \mathfrak{S}_{3\ell}$ uniformément au hasard et envoie $S_0^1||S_0^2||S_0^3||M_{100}||M_{010}||M_{001}||M_{000}||\sigma$ à $\mathcal{CM}$. Le PPTA $\mathcal{CM}$ possède donc

$$\Big(i_0, i_1, \big(k, N, \tau, (i, f_i)_i\big), \kappa, S_0^1||S_0^2||S_0^3||M_{100}||M_{010}||M_{001}||M_{000}||\sigma\Big)$$

pour déterminer β. Vu la distribution uniforme de S_0^1, S_0^2 et S_0^3, la probabilité que $\mathcal{CM}$ devine β est $\frac{1}{2}$, i.e. $Adv^{oa-comp}_{\mathcal{PAM}^{final},\mathcal{CM}}$ est nulle. □

Les messages entre le serveur d'authentification et les composantes malicieuses

Nous avions dit dans le paragraphe introductif en 4.3.4 que la confidentialité n'a pas pu être prouvée au niveau du serveur d'authentification pour $\mathcal{PAM}^{final}$. Nous expliquons le phénomène ici. Le problème se situe au niveau des messages qui sont envoyés par les bases de données au serveur d'authentification et apparaît à cause des masques choisis par le lecteur.

Dans l'étape **2)**, le lecteur $\mathcal{R}$ choisit huit masques $M_{b_1b_2b_3}$ de longueur k comme dans le cas où il y a huit vraies bases de données qui communiquent avec le serveur (voir le paragraphe 4.3.3) et les envoie aux bases de données $\mathcal{DB}_0$ et $\mathcal{DB}_1$. Il nous suffit de regarder ce qui se passe au niveau de $\mathcal{DB}_0$ pour comprendre où est le problème.

La composante $\mathcal{DB}_0$ reçoit de $\mathcal{R}$ (et après déchiffrement) la donnée

$$S_0^1||S_0^2||S_0^3||M_{000}||M_{100}||M_{010}||M_{001}||\sigma$$

où on rappelle que les S_0^i sont dans $\{1, ..., \ell\}$ et $\sigma \in \mathfrak{S}_{3\ell}$. Il calcule ensuite

$$X_0'' := X_{100}(1)||...||X_{100}(\ell)||X_{010}(1)||...||X_{010}(\ell)||X_{001}(1)||...||X_{001}(\ell)||X_{000}$$

où on rappelle que pour tout $t \in \{1, ..., \ell\}$ on a

$$X_{100}(t) := \bigoplus_{S_0^1 \oplus \{t\} \times S_0^2 \times S_0^3} f_{h_1h_2h_3},$$

$$X_{010}(t) := \bigoplus_{S_0^1 \times S_0^2 \oplus \{t\} \times S_0^3} f_{h_1h_2h_3},$$

et

$$X_{001}(t) := \bigoplus_{S_0^1 \times S_0^2 \times S_0^3 \oplus \{t\}} f_{h_1h_2h_3}$$

(d'après les spécifications du protocole de récupération de données utilisé). Puis il calcule

$$X_0' := M_{100}||...||M_{100}||M_{010}||...||M_{010}||M_{001}||...||M_{001}||M_{000} \oplus X_0''$$

et c'est ici que le problème apparaît. Pour tout $b_1b_2b_3 \in \{001, 010, 100\}$, l'un des $X_{b_1b_2b_3}(t)$ parmi les ℓ qui sont calculés est le vecteur que calculerait $\mathcal{DB}_{b_1b_2b_3}$ dans le cas où huit bases de données sont utilisées et dans ce cas le masque $M_{b_1b_2b_3}$ sert à cacher le résultat du serveur d'authentification. Or, pour que deux bases de données en émulent huit, il faut donner à $\mathcal{AS}$ **tous** les vecteurs $X_{b_1b_2b_3}(t)$, donc il faut tous les cacher. Comme $\mathcal{R}$ ne choisit que huit masques au départ, le même $M_{b_1b_2b_3}$ sert à cacher $X_{b_1b_2b_3}(t)$ pour tout t. C'est une mauvaise utilisation du chiffrement de Vernam. Ce problème n'apparaît pas dans le cas du protocole $\mathcal{PAM}^{semi-final}$ car le lecteur choisit non pas huit masques de longueur k mais (après que les bases de données aient fait la partition) $3\ell + 1$ masques de longueur k. C'est d'ailleurs ce qui le rend beaucoup moins efficace.

Un dernier calcul est fait par $\mathcal{DB}_0$ avant d'envoyer un message à $\mathcal{AS}$: il permute les 3ℓ premiers blocs de longueur k de X_0' selon σ pour obtenir ce qu'on note X_0. Cela ne change pas grand chose au problème.

Formellement, quand on essaye de prouver la sécurité contre un adversaire qui possède les vecteurs X_0 et X_1 (qui est calculé de la même manière que X_0 et qui souffre du même problème) on ne peut pas conclure *à priori* en disant simplement que ces vecteurs sont uniformément distribués car ce n'est clair du tout. Il faut faire une étude plus fine de la distribution des sommes de la forme

$$\bigoplus_{S^1 \oplus \{t\} \times S^2 \times S^3} f_{h_1h_2h_3},$$

$$\bigoplus_{S^1 \times S^2 \oplus \{t\} \times S^3} f_{h_1h_2h_3},$$

et

$$\bigoplus_{S^1 \times S^2 \times S^3 \oplus \{t\}} f_{h_1h_2h_3}$$

en sachant que les S sont uniformément distribués dans $\{1, ..., \ell\}$. Bien entendu, la distribution des $f_{h_1h_2h_3}$ dans la base de données joue un rôle aussi.

On conclut ce paragraphe en indiquant les adversaires concernés par cette faiblesse dans notre modèle de sécurité. Il s'agit du serveur d'authentification $\mathcal{AS}$ et des adversaires extérieurs. En ce qui concerne $\mathcal{AS}$, on ne peut encore rien conclure. Par contre, on peut contourner le problème pour des adversaires extérieurs : il suffit que $\mathcal{DB}_0$ et $\mathcal{DB}_1$ chiffrent X_0 et X_1 avant de les envoyer à $\mathcal{AS}$. C'est bien ce qu'on demande dans le protocole.

4.4 Quelques valeurs numériques

Ici nous indiquons quelques valeurs numériques obtenues lors de simulations (en langage C) de certains des protocoles. La réalisation complète de ces simu-

lations est toujours en cours.

4.4.1 Les protocoles final et semi-final

Dans les tableaux ci-dessous, on trouve la taille des données générées par les diverses composantes sans le chiffrement pour les protocoles dits "semi-final" et "final". On trouve aussi la taille de la base de données simulant celle des flux morphométriques. Toutes les données - à l'exception du nombre d'objets - sont en octets. Le premier tableau donne tous les messages excepté celui du lecteur vers les bases de données qui se trouve dans le deuxième. Ce dernier contient aussi le nombre d'octets que le lecteur doit choisir uniformément au hasard.

Ces données sont des fonctions de deux variables k_0 et ℓ_0 : les flux morphométriques sont de longueur $k = 4k_0$ octets et le nombre d'éléments dans la base de données est $N = \ell^3$ où $\ell = 32\ell_0$. La taille des messages et de la base de données est en octets.

- **Données communes aux protocoles final et semi-final**

k_0	ℓ_0	k	N	$\mathcal{DB}$	$\mathcal{R}$ vers $\mathcal{AS}$	$\mathcal{DB}_\beta$ vers $\mathcal{AS}$
1	1	4	32768	131072	16	388
4	1	16	32768	524288	28	1552
4	2	16	262144	4194304	28	3088
4	4	16	2097152	33554432	28	6160
8	2	32	262144	8388608	44	6176
8	4	32	2097152	67108864	44	12320

Envois de $\mathcal{R}$ vers $\mathcal{DB}$ et aléa produit par $\mathcal{R}$

		final	final	semi-final	semi-final
k_0	ℓ_0	$\mathcal{R}$ vers $\mathcal{DB}_\beta$	aléa produit par $\mathcal{R}$	$\mathcal{R}$ vers $\mathcal{DB}_\beta$	aléa produit par $\mathcal{R}$
1	1	412	32	784	776
4	1	460	128	1948	3104
4	2	856	128	3880	6176
4	4	1648	128	7744	12320
8	2	920	256	6968	12352
8	4	1712	256	13904	24640

4.4.2 Le protocole simple à deux bases de données

Les tableaux ci-dessous montrent les mêmes données en fonction de k_0 et ℓ_0 pour le protocole simple à deux bases de données. Ici, ℓ_0 est lié au nombre d'objets par la formule $N = 32\ell_0$ et $k = 4k_0$ est le nombre d'octets de chaque flux.

Les deux premiers tableaux montrent ce qu'on obtient si on prend les mêmes valeurs pour k_0 et ℓ_0 que dans les tableaux du paragraphe précédent. Cela donne notamment une bien plus faible quantité d'objets.

k_0	ℓ_0	k	N	$\mathcal{DB}$	$\mathcal{R}$ vers $\mathcal{AS}$	$\mathcal{DB}_\beta$ vers $\mathcal{AS}$
1	1	4	32	128	4	4
4	1	16	32	512	16	16
4	2	16	64	1024	16	16
4	4	16	128	2048	16	16
8	2	32	64	2048	32	32
8	4	32	128	4096	32	32

Envois de $\mathcal{R}$ vers $\mathcal{DB}$ et aléa produit par $\mathcal{R}$

		final	final
k_0	ℓ_0	$\mathcal{R}$ vers $\mathcal{DB}_\beta$	aléa produit par $\mathcal{R}$
1	1	8	12
4	1	20	36
4	2	32	48
4	4	48	64
8	2	48	80
8	4	64	96

Supposons maintenant qu'on veuille avoir un nombre d'objets comparable à celui obtenu dans les tableaux des protocoles final et semi-final et voyons la taille de l'aléa qu'il faut choisir. Comme ce qui importe le plus est la taille sous-ensemble choisi par le lecteur, c'est ℓ_0 qu'il importe de faire varier. On constate ici que l'aléa à choisir est extrêmement important pour s'autoriser à manipuler un grand nombre d'objets.

k_0	ℓ_0	k	N	$\mathcal{DB}$	$\mathcal{R}$ vers $\mathcal{AS}$	$\mathcal{DB}_\beta$ vers $\mathcal{AS}$
4	1024	16	32768	524288	16	16
4	8192	16	262144	4194304	16	16
4	65536	16	2097152	33554432	16	16

Envois de $\mathcal{R}$ vers $\mathcal{DB}$ et aléa produit par $\mathcal{R}$

k_0	ℓ_0	$\mathcal{R}$ vers $\mathcal{DB}_\beta$	aléa produit par $\mathcal{R}$
4	1024	4112	4128
4	8192	32784	32800
4	65536	262160	262176

On voit donc que le protocole simple à deux bases de données ne convient pas dans un système possèdant un lecteur faible s'il y a beaucoup d'objets à authentifier.

4.4.3 Le protocole utilisant El-Gamal bit-à-bit

Passons au protocole utilisant ElGamal bit-à-bit. Ici, le chiffrement est inclus dans les données. On fixe $k_0 = 4$ et donc $k = 16$, et on suppose qu'on travaille avec une courbe elliptique dont la taille du groupe sous-jacent est 224 bits, i.e. 28 octets. Enfin, le nombre d'objets est $N = 32\ell_0$.

ℓ_0	N	$\mathcal{DB}$	$\mathcal{R}$ vers $\mathcal{AS}$, $\mathcal{DB}$ vers $\mathcal{AS}$ et $\mathcal{AS}$ vers $\mathcal{M}$
1	32	512	3584
2	64	1024	3584
4	128	2048	3584

Envois de $\mathcal{R}$ vers $\mathcal{DB}$ et aléa produit par $\mathcal{R}$

ℓ_0	$\mathcal{R}$ vers $\mathcal{DB}$	aléa produit par $\mathcal{R}$
1	896	4480
2	1792	5376
4	3584	7168

Supposons maintenant qu'on veuille avoir un nombre d'objets comparable à celui obtenu dans les tableaux des protocoles final et semi-final et voyons la taille de l'aléa qu'il faut choisir. Comme ce qui importe le plus est la taille sous-ensemble choisi par le lecteur, c'est ℓ_0 qu'il importe de faire varier. On constate ici que l'aléa à choisir est extrêmement important pour s'autoriser à manipuler un grand nombre d'objets.

ℓ_0	N	$\mathcal{DB}$	$\mathcal{R}$ vers $\mathcal{AS}$, $\mathcal{DB}$ vers $\mathcal{AS}$ et $\mathcal{AS}$ vers $\mathcal{M}$
1024	32768	524288	3584
8192	262144	4194304	3584
65536	2097152	33554432	3584

Envois de $\mathcal{R}$ vers $\mathcal{DB}$ et aléa produit par $\mathcal{R}$

ℓ_0	$\mathcal{R}$ vers $\mathcal{DB}$	aléa produit par $\mathcal{R}$
1024	917504	921088
8192	7340032	7343616
65536	58720256	58723840

On en déduit que le protocole utilisant le chiffrement ElGamal bit-à-bit ne convient pas s'il y a beaucoup d'objets et un lecteur faible.

Conclusions

Nous avons construit quelques protocoles destinés à faire de l'authentification d'objets à distance en opérant en deux temps. Dans un premier temps, un flux morphométrique propre à l'objet en est extrait et est stocké dans une base de données à un indice qui est imprimé sur l'objet. Cette étape à lieu une seule fois avant la mise en circulation de l'objet. Ensuite, à chaque fois qu'une entité souhaite authentifier l'objet, un lecteur affilié au système extrait à nouveau un flux morphométrique de l'objet et l'envoie à un serveur ayant accès à la base de données stockée de manière à pouvoir comparer les flux. L'objet est considéré comme étant authentique si et seulement si ces flux sont suffisamment proches.

Les protocoles présentés ici sont de deux types essentiellement. L'un d'eux est une adaptation directe d'un protocole d'authentification biométrique imaginé par Bringer *et al.* dans [5] au cas des courbes elliptiques. Les autres sont des protocoles qui peuvent être implémentés avec n'importe quel algorithme de chiffrement à clef publique pourvu que ce dernier soit au moins sémantiquement sécurisé, et qui sont construits également avec un protocole de récupération de données décrit dans [9].

Nos protocoles sont presque tous prouvés sécurisés dans le modèle de sécurité utilisé, en utilisant une nouvelle caractérisation de la sécurité sémantique. On cherche à se prémunir contre des adversaires extérieurs au système qui essayent de modifier les communications entre le lecteur et le serveur afin de changer la réponse finale, ainsi que contre des adversaires extérieurs et intérieurs au système qui cherchent à déterminer quel objet est testé à priori. Seule la confidentialité au niveau de la composante $\mathcal{AS}$ (serveur d'authentification) du protocole dit final n'a pu encore être démontrée, mais ce n'est pas le point de sécurité le plus sensible.

On peut imaginer que la base de données contienne jusqu'à 10^9 objets, par exemple s'il s'agit d'authentifier des passeports ou des produits en série. Il résulte des données numériques que les seuls protocoles adaptés à une telle quantité d'objets sont les protocoles dits final et semi-final qui permettent au lecteur d'avoir une quantité raisonnable d'octets à choisir de manière pseudo-aléatoire par rapport à la taille de la base de données. Le protocole simple à deux bases de données ou celui utilisant le chiffrement ElGamal bit-à-bit ne sont intéressants que pour de très petits nombres d'objets ou des lecteurs extrêmement puissants.

Parmi les pistes de recherche à suivre se trouvent d'abord le problème de

rendre ces protocoles plus performants en utilisant d'autres protocoles de récupération de données notamment. Il serait aussi intéressant de renforcer le modèle de sécurité en étendant l'exigence d'intégrité à toutes les communications entre les composantes. Enfin, il reste à étudier plus en détail la sécurité au niveau du serveur d'authentification du protocole final.

Publications

Article publié : Jean Lancrenon, Roland Gillard, Thierry Fournel, "Remote object authentication : confidence model, cryptosystem, and protocol", dans *Data Mining, Intrusion Detection, Information Security and Assurance, and Data Network Security 2009*, ed. Belur V. Dasarathy, Proceedings of SPIE Vol. 7344, 20, 10 pages

Article en cours de publication : Jean Lancrenon, Roland Gillard, Thierry Fournel, "Remote object authentication against counterfeiting using elliptic curves", qui sera présenté à la conférence *SPIE Defense, Security, and Sensing*, 25-29 April 2011 in Orlando, Florida Etats-Unis d'Amérique, 10 pages

Article en préparation : "Isolating partial information of indistinguishable encryptions" (titre provisoire) pour les résultats de sécurité théorique

Bibliographie

[1] Bellare, M., Boldyreva, A., et Micali, S., *Public-key encryption in a multi-user setting : security proofs and improvements*, Advances in cryptology - EUROCRYPT 2000, LNCS Vol. 1807, pp. 259-274, Springer, 2000.

[2] Blake, I.F., Seroussi, G., Smart, N., *Elliptic Curves in Cryptography*, London Mathematical Society Lecture Notes Series, 265, Cambridge University Press, 1999

[3] Blake, I.F., Seroussi, G., Smart, N., *Advances in Elliptic Curve Cryptography*, London Mathematical Society Lecture Notes Series, 317, Cambridge University Press, 2005

[4] Bolle, R.M., Connel, J.H., Pankanti, S., Ratha, N.K., Senior, A.W., *Guide to biometrics*, Springer Verlag, New York, 2003

[5] Bringer, J., Chabanne, H., Izabachène, M., Pointcheval, D., Tang, Q., Zimmer, S., *An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication*, Proceedings of ACISP '07, J. Pieprzyk, H. Ghodosi and E. Dawson Eds., LNCS 4586, Springer-Verlag, pp. 96-106, 2007

[6] Bringer, J., Chabanne, H., *An authentication protocol with encrypted biometric data*, Progress in cryptology, AFRICACRYPT 2008, LNCS 5023, Springer, Heidelberg, pp. 109-124, 2008

[7] Bringer, J., Chabanne, H., Pointcheval, D., Tang, Q., *Extended private information retrieval and its application in biometrics authentication*, Cryptology and Network Security, LNCS 4856, Springer, pp. 175-193, 2007

[8] Boneh, D., *The decision Diffie-Hellman problem*, Proceedings of the Third Algorithmic Number Theory Symposium, LNCS 1423, Springer, pp. 48-63, 1998

[9] Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M., *Private Information Retrieval*, Proceedings of the 36th annual IEEE Conference on Foundations of Computer Science (Oct.). IEEE, New York, pp. 41-50, 1995

[10] ElGamal, T., *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Trans. Inform. Theory 31, pp. 469-472, 1985

[11] Fournel T., Coltuc D., Becker J.-M., Boutant Y., *Multiscale extraction of uncompressive bitstrings from speckle patterns*, Proc. Workshop on Information Optics 2008, JPCS, 139, 2008

[12] Fournel, T., Gillard, R., Becker, J.-M., Boutant, Y., *Morpho-cryptography : a new way for securing both information and storage media*, Proc. SAR-SSI 2007, Annecy, 12-15 juin 2007

[13] Frey, G., Rück, H., *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, Mathematics of Computation 62, pp. 865-874, 1994

[14] Gaudry, P., Hess, F., Smart, N., *Constructive and destructive facets of Weil descent on elliptici curves*, Journal of Cryptology vol. 15, pp. 19-34, 2002

[15] Goldreich, O., *Foundations of cryptography I : basic tools*, Cambridge University Press, 2001.

[16] Goldreich, O., *Foundations of cryptography II : basic applications*, Cambridge University Press, 2004.

[17] Goldwasser, S., Micali, S., *Probabilistic encryption and how to play mental poker keeping secret all partial information*, Proceedings of the 14th annual ACM Symposium on theory of Computing, San Francisco, ACM, New York, pp. 365-377, 1982

[18] Gordon, D., *Discrete logarithms in GF(p) using the number field sieve*, SIAM Journal on Discrete Mathematics 6, pp. 124-138, 1993

[19] Hess, F., *Generalizing the GHS attack on the elliptic curve discrete logarithm problem*, LMS Journal of Computation and Mathematics vol. 7, pp. 167-192, 2004

[20] Hankerson, D., Menezes, A., Vanstone, S., *Guide to elliptic curve cryptography*, Springer, December 2003

[21] Joux, A., Nguyen, K., *Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups*, Journal of Cryptology vol. 16, pp. 239-247, 2003

[22] Koblitz, N., *Elliptic curve cryptosystems*, Mathematics of Computation 48, pp. 203-209, 1987

[23] Elliptic curve cryptography : the serpentine course of a paradigm shift, Journal of Number Theory, à paraître

[24] Kushilevitz, E., Ostrovsky, R., *Replication is not needed : single database, computationally-private information retrieval*, 38th FOCS, pp. 364-373, 1997

[25] Lenstra, H.W., Jr., *Factoring integers with elliptic curves*, Annals of Mathematics 126, pp. 649-673, 1987

[26] Lenstra, A., Lenstra, H., *The developement of the number field sieve*, Springer, 1993

[27] Lipmaa, H., *An oblivious transfer protocol with log-squared communications*, ISC 2005, LNCS 3650, Springer, Heidelberg, pp. 314-328, 2005

[28] Menezes, A., Okamoto, T., Vanstone, S., *Reducing elliptic curve logarithms to logarithms in a finite field*, IEEE Trans. Inform. Theory 39, 1639-1646, 1993

[29] Miller, V., *Use of elliptic curves in cryptography*, CRYPTO 85, 1985

[30] Pappu R., Recht B., Taylor J., Gershenfeld N., *Physical one-way functions*, Science, 297, pp. 2026-2030, 2002

[31] Pollard, J., *Factoring with cubic integers*, The development of the number field sieve, ed. Lenstra, A. et Lenstra, H., pp. 4-10, Springer, 1993

[32] Pollard, J., *Theorems on factorization and primality testing*, Proc. Cambridge Philos. Soc. 76, pp. 521-528, 1974

[33] Rothe, J., *Complexity theory and cryptology : an introduction to cryptocomplexity*, Springer, 2005

[34] Shoup, V., *Sequences of games : a tool for taming complexity in security proofs*, manuscript, 2006

[35] Silverman, J., *The arithmetic of ellipitic curves*, GTM 106, Springer, 1986

[36] Smart, N., *The discrete logarithm problem on elliptic curves of trace 1*, Journal of Cryptology vol. 12, pp. 193-196, 1999

[37] Talbot, J., Welsh, D., *Complexity and cryptography : an introduction*, Cambridge University Press, 2006

[38] Tsiounis, Y., Yung, M., *On the security of ElGamal-based encryption*, Proceedings of the First Annual Workshop on Practice and Theory in Public-Key Crypotography : Public Key Cryptography, LNCS 1431, pp. 117-134, 1998

Résumé

Cette thèse est consacrée à la description et à l'étude de la sécurité de divers protocoles destinés à faire de l'authentification d'objets physiques à distance. Nous partons de la possibilité d'extraire d'un objet un flux de données qui lui est propre codé en binaire. Cette extraction est faite une fois avant la mise en circulation de l'objet, et le résultat est stocké dans une base de données sécurisée se trouvant dans un serveur central. Puis, à chaque fois que cela s'avère nécessaire, un lecteur mobile communicant avec le serveur central extrait de l'objet un autre exemplaire de ce flux de données et l'envoie au serveur pour faire une comparaison.

L'objectif des protocoles proposés est de pouvoir réaliser une authentification en garantissant d'une part que les informations envoyées et reçues par le lecteur n'ont pas été manipulées par un adversaire extérieur et d'autre part sans révéler l'identité de l'objet testé à un tel adversaire, ou même, modulo certaines hypothèses raisonnables, aux composantes du système. Nous nous sommes fixés de plus comme objectif d'utiliser des méthodes de cryptographie sur courbe elliptique pour pouvoir profiter des bonnes propriétés de ces dernières, notamment une sécurité accrue par rapport à la taille des clefs utilisées.

Nous présentons plusieurs protocoles atteignant l'objectif et établissons pour presque tous une preuve théorique de leur sécurité. Pour ce faire, il nous a été nécessaire de développer une nouvelle caractérisation d'une notion standard de sécurité. Nous terminons par un tableau comparatif de performances pour les protocoles qui ont été simulés.

Mots-clefs

Protocole d'authentification, cryptosystème, modèle de sécurité, authentification d'objets, traçabilité, sécurité sémantique, chiffrements indistinguables, informations partielles