Design and implementation of lightweight and secure cryptographic algorithms for embedded devices

Lama Sleem

UBFC
UNIVERSITÉ
BOURGOGNE FRANCHE-COMTÉ

UFC
UNIVERSITÉ
DE FRANCHE-COMTÉ

THÈSE DE DOCTORAT DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

LAMA SLEEM

Design and implementation of lightweight and secure cryptographic algorithms for embedded devices

Conception et implémentation d'algorithmes cryptographiques légers et sécurisés pour dispositifs embarqués

Thèse présentée et soutenue à Belfort, le 17 Janvier 2020

Composition du Jury :

OUSSAMA BAZZI PROFESSEUR	UNIVERISTE LIBANAISE	Rapporteur
PIERRE SPITERI PROFESSEUR	ENSEEIHT	Rapporteur
CHRISTOPHE GUYEUX PROFESSEUR	Université Bourgogne Franche-Comté	Examinateur
FLAVIEN VERNIER MAÎTRE DE CONFÉRENCES	Université Savoie Mont Blanc	Examinateur
RAPHAËL COUTURIER PROFESSEUR	Université Bourgogne Franche-Comté	Directeur de thèse

N° | X | X | X |

école doctorale **sciences pour l'ingénieur et microtechniques**

Title: Design and implementation of lightweight and secure cryptographic algorithms for embedded devices

Keywords: Image encryption, Encryption, Decryption, Confusion, Diffusion, dynamic key, Security tests.

Abstract:

Living in an era where new devices are astonishing considering their high capabilities, new visions and terms have emerged. Moving to smart phones, Wireless Sensor Networks, high-resolution cameras, pads and much more, has mandated the need to rethink the technological strategy that is used today. Starting from social media, where apparently everything is being exposed, moving to highly powerful surveillance cameras, in addition to real time health monitoring, it can be seen that a high amount of data is being stored in the Cloud and servers. This introduced a great challenge for their storage and transmission especially in the limited resourced platforms that are characterized by: (a) limited computing capabilities, (b) limited energy and source of power and (c) open infrastructures that transmit data over wireless unreliable networks. One of the extensively studied platforms is the Vehicular Ad-hoc Networks which tends to have many limitations concerning the security field.
In this dissertation, we focus on improving the security of transmitted multimedia contents in different limited platforms, while preserving a high security level. Limitations of these platforms are taken into consideration while enhancing the execution time of the secure cipher. Additionally, if the proposed cipher is to be used for images, the intrinsic voluminous and complex nature of the managed images is also taken into account.
In the first part, we surveyed one of the limited platforms that is interesting for many researchers, which is the Vehicular Ad-hoc Networks. In order to pave the way for researchers to find new efficient security solutions, it is important to have one reference that can sum most of the recent works. It almost investigates every aspect in this field shedding the light over different aspects this platform possesses.
Then, in order to propose any new security solution and validate its robustness and the level of randomness of the ciphered image, a simple and efficient test is proposed. This test proposes using the randomness tools, TestU01 and Practrand, in order to assure a high level of randomness. After running these tests on well known ciphers, some flaws were exposed.
Proceeding to the next part, a novel proposal for enhancing the well-known ultra lightweight cipher scheme, **Speck**, is proposed. The main contribution of this work is to obtain a better version compared to Speck. In this proposal, 26 rounds in Speck were reduced to 7 rounds in **Speck-R** while enhancing the execution time by at least 50%. First, we validate that Speck-R meets the randomness tests that are previously proposed. Additionally, a dynamic substitution layer adds more security against key related attacks and highly fortifies the cipher. Speck-R was implemented on different limited arduino chips and in all cases, Speck-R was ahead of Speck.
Then, in order to prove that this cipher can be used for securing images, especially in VANETS/IoV, where images can be extensively re/transmitted, several tests were exerted and results showed that Speck-R indeed possesses the high level of security desired in any trusted cipher. Extensive experiments validate our proposal from both security and performance point of views and demonstrate the robustness of the proposed scheme against the most-known types of attacks.

Titre : Design and implementation of lightweight and secure cryptographic algorithms for embedded devices

Mots-clés : Chiffrement d'images, Chiffrement, Déchiffrement, Confusion, Diffusion, dynamique Tests de sécurité.

Résumé :

Nous vivons actuellement dans une ère avec sans cesse de nouveaux appareils technologiques (smartphone, réseaux de capteurs sans fil, aux caméras haute résolution, etc). En partant des médias sociaux, en passant par des caméras de surveillance très puissantes, et sans oublier la surveillance de la santé en temps réel, on constate qu'une grande quantité de données est stockée dans le cloud et les serveurs. Cela représente un grand défi de stockage et de transmission, en particulier dans les plates-formes aux ressources limitées qui sont caractérisées par : (a) des capacités de calcul limitées, (b) une source d'énergie limitées et (c) des infrastructures ouvertes qui transmettent des données sur des réseaux sans fil peu fiables. Dans cette thèse, nous nous concentrons sur l'amélioration de la sécurité des contenus multimédia transmis sur des plates-formes à capacité de calcul limitée, tout en préservant un niveau de sécurité élevé. Dans la première partie, nous avons étudié les réseaux ad hoc véhiculaire. Nous avons proposé un état de l'art qui permet de résumer la plupart des travaux récents et d'explorer presque tous les aspects de ce domaine en illustrant les différents aspects que possède cette plateforme. Ensuite, afin de proposer une nouvelle solution de sécurité et de valider sa robustesse et le niveau de caractère aléatoire d'une image chiffrée, nous avons proposé un test simple et efficace. Celui-ci est basé sur des outils pour tester statistiquement le caractère aléatoire de nombres pseudo aléatoires, TestU01 et Practrand. Après avoir effectué ces tests sur des algorithmes de chiffrement bien connus, certaines failles ont été exposées et une nouvelle proposition visant à améliorer le système de chiffrement ultra-léger Speck est proposée. La principale contribution de ce travail est d'obtenir une meilleure version par rapport à Speck. Dans cette nouvelle proposition, appelée Speck-R, nous utilisons seulement 7 itérations contrairement à Speck qui en utilise 26 et nous réduisons le temps d'exécution d'au moins 50%. Tout d'abord, nous validons que Speck-R répond aux tests de statistiques pour mesurer l'aléatoire, proposés précédemment. De plus, nous avons rajouté un système de clé dynamique qui procure plus de sécurité contre les attaques liées à la clé. Speck-R a été implémenté sur différentes cartes de type arduino et dans tous les cas, Speck-R était plus rapide que Speck. Ensuite, afin de prouver que ce chiffrement peut être utilisé pour sécuriser les images, en particulier dans les réseaux VANETS/IoV, plusieurs tests ont été effectués et les résultats montrent que Speck-R possède effectivement le haut niveau de sécurité souhaité. Des expérimentations valident notre proposition du point de vue de la sécurité et de la performance et démontrent la robustesse du système proposé face aux types d'attaques les plus connus.

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besanon, France

ABSTRACT

Design and implementation of lightweight and secure cryptographic algorithms for embedded devices

Lama SLEEM
University of Bourgogne Franche Comté, 2020

Supervisor:Raphaël COUTURIER

Living in an era where new devices are astonishing considering their high capabilities, new visions and terms have emerged. Moving to smart phones, Wireless Sensor Networks, high-resolution cameras, pads and much more, has mandated the need to rethink the technological strategy that is used today. Starting from social media, where apparently everything is being exposed, moving to highly powerful surveillance cameras, in addition to real time health monitoring, it can be seen that a high amount of data is being stored in the Cloud and servers. This introduced a great challenge for their storage and transmission especially in the limited resourced platforms that are characterized by: (a) limited computing capabilities, (b) limited energy and source of power and (c) open infrastructures that transmit data over wireless unreliable networks. One of the extensively studied platforms is the Vehicular Ad-hoc Networks which tends to have many limitations concerning the security field.

In this dissertation, we focus on improving the security of transmitted multimedia contents in different limited platforms, while preserving a high security level. Limitations of these platforms are taken into consideration while enhancing the execution time of the secure cipher. Additionally, if the proposed cipher is to be used for images, the intrinsic voluminous and complex nature of the managed images is also taken into account.

In the first part, we surveyed one of the limited platforms that is interesting for many researchers, which is the Vehicular Ad-hoc Networks. In order to pave the way for researchers to find new efficient security solutions, it is important to have one reference that can sum most of the recent works. It almost investigates every aspect in this field shedding the light over different aspects this platform possesses.

Then, in order to propose any new security solution and validate its robustness and the level of randomness of the ciphered image, a simple and efficient test is proposed. This test proposes using the randomness tools, TestU01 and Practrand, in order to assure a high level of randomness. After running these tests on well known ciphers, some flaws were exposed.

Proceeding to the next part, a novel proposal for enhancing the well-known ultra lightweight cipher scheme, **Speck**, is proposed. The main contribution of this work is

to obtain a better version compared to Speck. In this proposal, 26 rounds in Speck were reduced to 7 rounds in **Speck-R** while enhancing the execution time by at least 50%. First, we validate that Speck-R meets the randomness tests that are previously proposed. Additionally, a dynamic substitution layer adds more security against key related attacks and highly fortifies the cipher. Speck-R was implemented on different limited arduino chips and in all cases, Speck-R was ahead of Speck.

Then, in order to prove that this cipher can be used for securing images, especially in VANETS/IoV, where images can be extensively re/transmitted, several tests were exerted and results showed that Speck-R indeed possesses the high level of security desired in any trusted cipher. Extensive experiments validate our proposal from both security and performance point of views and demonstrate the robustness of the proposed scheme against the most-known types of attacks.

RÉSUMÉ

Conception et implémentation d'algorithmes cryptographiques légers et sécurisés pour dispositifs embarqués

Lama SLEEM
University of Bourgogne Franche Comté, 2020

Supervisor:Raphaël COUTURIER

Nous vivons actuellement dans une ère avec sans cesse de nouveaux appareils technologiques (smartphone, réseaux de capteurs sans fil, aux caméras haute résolution, etc). En partant des médias sociaux, en passant par des caméras de surveillance très puissantes, et sans oublier la surveillance de la santé en temps réel, on constate qu'une grande quantité de données est stockée dans le cloud et les serveurs. Cela représente un grand défi de stockage et de transmission, en particulier dans les plates-formes aux ressources limitées qui sont caractérisées par : (a) des capacités de calcul limitées, (b) une source d'énergie limitées et (c) des infrastructures ouvertes qui transmettent des données sur des réseaux sans fil peu fiables. Dans cette thèse, nous nous concentrons sur l'amélioration de la sécurité des contenus multimédia transmis sur des plates-formes à capacité de calcul limitée, tout en préservant un niveau de sécurité élevé. Dans la première partie, nous avons étudié les réseaux ad hoc véhiculaire. Nous avons proposé un état de l'art qui permet de résumer la plupart des travaux récents et d'explorer presque tous les aspects de ce domaine en illustrant les différents aspects que possède cette plateforme. Ensuite, afin de proposer une nouvelle solution de sécurité et de valider sa robustesse et le niveau de caractère aléatoire d'une image chiffrée, nous avons proposé un test simple et efficace. Celui-ci est basé sur des outils pour tester statistiquement le caractère aléatoire de nombres pseudo aléatoires, TestU01 et Practrand. Après avoir effectué ces tests sur des algorithmes de chiffrement bien connus, certaines failles ont été exposées et une nouvelle proposition visant à améliorer le système de chiffrement ultra-léger Speck est proposée. La principale contribution de ce travail est d'obtenir une meilleure version par rapport à Speck. Dans cette nouvelle proposition, appelée Speck-R, nous utilisons seulement 7 itérations contrairement à Speck qui en utilise 26 et nous réduisons le temps d'exécution d'au moins 50%. Tout d'abord, nous validons que Speck-R répond aux tests de statistiques pour mesurer l'aléatoire, proposés précédemment. De plus, nous avons rajouté un système de clé dynamique qui procure plus de sécurité contre les attaques liées à la clé. Speck-R a été implémenté sur différentes cartes de type arduino et dans tous les cas, Speck-R était plus rapide que Speck. Ensuite, afin de prouver que ce chiffrement peut être utilisé pour sécuriser les images, en particulier dans les réseaux VANETS/IoV, plusieurs tests ont été effectués et les résultats montrent que Speck-R possède effectivement le haut niveau de sécurité souhaité. Des

expérimentations valident notre proposition du point de vue de la sécurité et de la performance et démontrent la robustesse du système proposé face aux types d'attaques les plus connus.

CONTENTS

List of Figures

List of Tables

LIST OF ALGORITHMS

List of Definitions

ABBREVIATIONS

DEDICATION

”There are those who give with joy, and that joy is their reward.” Khalil Gibran

** This dissertation is dedicated to *My Parents* Abbas and Ibtissam who instilled in me the virtues of perseverance and commitment and relentlessly encouraged me to strive for excellence. Thank you Dad for all your support during my journey, you were and will always be my strength to chase my dreams.

** I dedicate this thesis to my supervisor Raphaël Couturier, nothing would have been possible without your patience, faith and great support.

** For my amazing brothers Fadl and Bilal, and my precious sister Ghina I dedicate this thesis for you and I thank you for all the support you gave me during this journey.

**For the one who was beside me at all the good times and at the bad ones, Ali, this work is dedicated to you, thank you for being my strength.

** I dedicate this work to all people who might find the work interesting and bring it on. It is for those people that this work has been written, and to them, this is dedicated.

Finally, always remember that **”It always seems impossible, until it’s done”-Nelson Mandela**

ACKNOWLEDGMENTS

First of all, I want to express my gratitude for the establishment ”Université Bourgogne-Franche-Comté”, for giving me the chance to fulfill my studies especially Prof. Thérèse Leblois.
I would like to express my gratitude for my supervisor Prof. Raphaël COUTURIER. You are such an amazing person who distributes joy, immense knowledge, positive energy and faith in the lab. Thank you for all the efforts you exerted and for all the advice, knowledge and strength you gave me. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so patiently and promptly.
Credits for part of the work go to the ”Mésocentre de calcul de Franche-Comté”. Thank you for giving us access to your machines that allowed us to have better and faster results.
Besides my supervisors, I would like to express my deep gratitude to Oussama BAZZI and Pierre SPITERI for accepting to review my dissertation. I would want to thank also Christophe GUYEUX and Flavien VERNIER for accepting to participate in my dissertation committee and for their appreciated comments.
My appreciation goes to the AND team (Algorithmique Numerique Distribuee): Abdallah Makhoul, Jean-François Couchot, Stephane Domas, Mourad Hakem, David Laiymani, Gilles Perrot, David Laiymani and Jean-Claude Charr. It has been a pleasure to know you all.
A special gratitude goes to my friend Nisreen Khernane, for all the support, the strength and all the efforts she exerted. I truly thank you and will never be able to forget how much impact you left deep inside me.
My appreciation goes also to my friends in the lab, whom I am so proud to know. Gaby, Joseph, Christian, Carol, Rania, Nancy, Ibrahim, Hicham, Anthony, Ralph, Héber, and Zhi Hao, I wish you all the success in the future, .
I would like to thank my friends outside the lab, who were my family in Belfort, Marwa, Ghadir and Joelle, you are more than sisters to me and this thesis is dedicated for you. Will never forget the moments we passed together having each others' backs.
My friends in Lebanon, Ali Hamoud, Kholoud, Mouhamad Assaf, Leen, Hussien M. and Mahdi, although we are far away now, but I carry you in my heart wherever I go.

My family, Dad, Mom, my brothers and my sister, we only know that great achievements deserve great sacrifices and you have sacrificed a lot. No thank you can pay you off your deeds. I simply say how lucky I am to have you as my precious family. For Reem, I am so lucky to be aunt.
Last but not least, I want to thank the person who were there from the very first beginning, pushing me and encouraging me to the best version of who I am, Ali, I thank God every day for finding someone like you. We will always be there for each other.
I do not dare to forget anyone, so I want to thank you all for whoever was there for me.

1

INTRODUCTION

1

INTRODUCTION

”If you reveal your secrets to the wind, you should not blame the wind for revealing them to the trees -”Khalil Gibran

1.1/ GENERAL INTRODUCTION

Security is no longer an option in the new technological invasion we are witnessing today. In 2019, a list of data breaches and cyber attacks took place reaching a 114.6 million records leakage as for August 2019 [380]. Whats app [381], Facebook [372], PDF (Portable Document Format) [384] were all exposed due to vulnerabilities in their platforms. Security has become one of the most important fields that interests both industrial and research organizations. Our data is being spread and stored on Servers, Clouds, and in billions of different places. All kinds of data (videos, images, texts, etc...) are being exchanged among users in all kinds of channels, Wireless, Satellites etc...

The increased size of the data exchanged has raised the necessity to find new security solutions that adapt to this enormous level of change. Many researchers are proposing new security solutions and extensive tests are being exerted to prove their points of view. However, many of these researchers are not taking into account the new emerging platforms that mandate specific criteria. For example, Wireless Sensor Networks (WSN), Internet of Things, Surveillance, Internet of Vehicles (IoV) etc... are all new platforms that have new requirements. These platforms need new security solutions that are suitable for the limited abilities they own. For example, in WSNs and IoT the devices used are limited in terms of battery, memory and computational power. In IoV the devices used have high mobility and need a fast algorithm to go along with the existing alterations. Furthermore, most of time communication failures often occur since the nodes use unreliable wireless communication to form a wide network. These intrinsic hardware constraints of the devices, in addition to multimedia data (i.e., large volume, real-time delivery and content richness), command new theoretical and practical challenges on the design and the development of such platforms. All these requirements motivate the need of developing integrated approaches that would consider all these constraints at once.

Moreover, encryption should be convenient to any kind of data, which means that the cipher used should be able to secure images, texts, and videos. Many ciphers do not take into consideration the intrinsic features that images and videos hold. Therefore, they

can succeed in securing texts, but many of them fail to meet the required level in other kinds of data.

Reaching the desired level of security, preserving the resources and resiliency against attacks are the main concerns of this thesis. Mainly, three important aspects need to be ensured:

- High level of security: In order to cope with the new attacks that are available today, ensuring a high level of security is our sole aim. Preserving the security of any kind of transmitted data is the basic motivation in this work.
- Low computational complexity: In order to adapt to the new demanding limitations, the proposed solutions need to possess a low level of computational complexity. Heavy operations, and exhausting processes should be eliminated to face the memory and power restrictions.
- Low error propagation: Since data is transmitted in unreliable noisy channels known as noisy environments [201], which introduce errors and toggle some bits within data, the receiver must be able to extract the content of the transmitted data, even when the latter has been corrupted by some perturbations.

In this context, this thesis tries to take all the missing links to deploy new security solutions that will secure any contents of data, respecting the limitations that these platforms suffer from. Having a security solution that is capable of excelling in reaching a high level of security as well as preserving the resources of the infrastructure, will be of great advantage.

1.2/ MAIN CONTRIBUTIONS OF THIS DISSERTATION

The main contributions of this dissertation concentrate on reaching a secure solution that adapts to the new constraints available in new platforms and new modern applications. Software and hardware efficiency should be of equal importance to reach an optimal security solution. To fulfill the previously mentioned requirements, we summarize the following contributions of this dissertation:

1. Two of the platforms that are under extensive security research are the Vehicular Ad-Hoc Network/VANET and the Internet of Vehicles/ IoV. These two platforms have many limitations that form many obstacles to obtain a safe and secure platform. In this work, and in order to pave the way for other researchers to find new suitable security solutions to these platforms, we surveyed both IoV and VANET, stating the difference between them especially in terms of security. The main contribution of this work can be divided into two major points. First, since VANET evolved into the IoV, it is important to indicate the differences between these two platforms and what are the main motivations behind this evolution. The second contribution is related to the security of such systems. Whether talking about VANET or IoV, using ad-hoc wireless communications, or 4G/5G communications, these platforms are

sensitive to a large number of threats. Unreliable multi-hop transmissions, willful intermediate packet forwarding, and sharing specific personal data (location information or any sensitive messages) will certainly require a specific level of security. Therefore, the practical benefits of VANET/IoV could be mitigated in the absence of appropriate safety systems and could have a negative reverse effect on traffic and drivers' safety. Thus, the main requirement to ensure is the security of these heterogeneous systems where several security requirements are needed to resist different kinds of existing attacks. Shortly, this chapter summarizes and studies the various cryptographic/non-cryptographic schemes that have been separately proposed to secure VANET/IoV. Moreover, the existing security challenges for these schemes are well presented. **In opposition to several existing works, almost all recent existing primitive security solutions are analyzed for each threat jeopardizing the safety of these platforms.** Each attack usually aims at affecting at least one of the following security services: availability, authentication, integrity, confidentiality, privacy, and non-reputation. **Attacks are classified according to their security impact as well as the corresponding layer(s) in the Internet protocol stack layer they could affect**. In fact, a better classification of the existing attacks, threats on different network layers, and their countermeasures would allow researchers to find new and more effective security measures.

2. After classifying threats, attacks and studying the impact of each attack on each layer, we propose a new technique to study the randomness for any proposed cipher in any platform. Creating randomness in the ciphered output and the importance of securing the exchanged information is the basic necessity in any new/old proposed cipher. Many tools are used to classify the efficiency of a security algorithm to assure the robustness and the validity. In this thesis, we propose using the famous tools TestU01 and Practrand, that are mainly used for pseudo-random generators, to validate the randomness of any ciphered output. These tools are available for free and are easy to be implemented and used. Finding the randomness faults in any cipher is the first step to enhance the security it offers. To prove our point of view, we used these tools to evaluate some of the famous encryption algorithms. Using different cryptographic libraries or codes from famous coders, we found that some of the tested algorithms fail the tests undertaken. These tests expose the randomness faults in these algorithms and highlight the codes that possess different vulnerabilities. In this work, we conclude that whenever a researcher wants to propose a new cipher, we propose using these tools that will either assure his proposal or weaken it. Moreover, using different codes from different libraries, we showed that the same algorithm may have some vulnerabilities in some cryptographic libraries while in others it does not.

3. As a next step, we propose a new cipher scheme that is an ultra-lightweight cipher, Speck-R where the "R" stands for Reduced. This cipher is based on the famous cipher Speck that comes to serve mostly the constrained and limited devices. After investigating two of the newest platforms, VANET and IoV, we need a cipher that can adapt to these limitations available in this kind of systems. Speck-R is a flexible, lightweight, robust and a dynamic cipher. Dynamicity here means that according to a dynamic key generated in a secure way, the cipher primitives will change. Using a dynamic approach will provide the cipher with more robustness and immunity against different kinds of attacks. Speck-R has the same features of Speck, but with a **R**educed number of rounds. By adding a dynamic substitution layer, we were

able to decrease the number of rounds from 26 to 7. The number of rounds here is based on using a 96 bits key with a block size of 64 bits. This proposal can be applied for other versions of Speck, since Speck has many versions using different key size and block size. Reaching a lower number of rounds is the first objective reached in this proposal. After that, and to prove that this cipher has the desired randomness and security, we used TestU01 and Practrand to prove our point of view. All the randomness tests were passed and thus we can say that according to our proposal, this cipher meets the desired security. Other tools were also used to validate our proposal like SET tool that is used to show the robustness of the substitution layer. Then, as mentioned previously, and to prove that this cipher can be used for different kinds of data, especially for images, Speck-R undergoes many tests to prove its reliability for image encryption as well. All the tests done render this cipher robust for both images and texts encryption. Finally, and to show the effectiveness of Speck-R in terms of hardware implementation, it was implemented on small chips that have limited power and are mainly used for testing in the Internet of Things environment. The results showed that Speck-R stands ahead of Speck in the three different tested chips. The enhancement in the execution time is at least 18.34% to reach a maximum of 77% depending on the chip used. To end, this cipher validated our previous proposals and can be used in the investigated platforms that have many limitations and challenges.

1.3/ DISSERTATION OUTLINE

The dissertation is organized as follows: Chapter 2 presents the scientific background about cryptography in general listing almost all the basic foundations needed. Then, Chapter 3 lists the proposed lightweight cryptographic solutions that have been proposed to the current date. Then, Chapter 4 presents the first contribution which is a survey done to differentiate between two of the newest platforms Vehicular Ad-Hoc Network and Internet of Vehicles, along with classifying the security threats in these new systems. The second contribution in Chapter 5 is a new proposal to validate the security and the randomness for any cipher in any platform. This proposal suggests using TestU01 and Practrand to validate the randomness of the ciphered output and it is validated by running it onto different famous ciphers. Then, in Chapter 6, the fourth contribution is explained which is a new cryptographic cipher based on the original cipher Speck. The new cipher is called Speck-R which is considered to be an ultra-lightweight cipher that can be used in limited and constrained devices. Decreasing the number of rounds from 26 to 7 is the most important achievement in this chapter. However, to prove the high security level of this cipher, many tests were launched and the obtained results back-boned the proposal. Finally, Chapter 7 gives a brief conclusion about the whole work presented in this thesis.

II

SCIENTIFIC BACKGROUND

2

A Brief Introduction to the History of Cryptography

There exists an expression in Latin: ”scientia potentia est”, which is translated into ”knowledge is power”, and assures how much knowledge is essential to us, humans. However, on the other side it means that people are also clung to the idea of protecting their information and any leakage in this data, will grant this knowledge-power to other parties. Here comes the idea of **Cryptology** originating from the Greek word kryptós which means hidden or secret. In fact, the term Cryptology was first used in 1935 according to [199]. It is the science that studies communication and storage of data in secure and usually secret form. Cryptology underpins both **a)** cryptography and **b)** cryptanalysis. Cryptography is derived from the Greek gráphein which is to write. It was first mentioned in 1658 and was originally the study of the principles and techniques by which information could be hidden using ciphers and revealed only by the legitimate users employing the shared secret key. However, now it targets the whole area of key-controlled transformations of information into forms that are impossible and/or computationally incapable for the third parties to duplicate or undo. For the cryptanalysis, it comes from the Greek analýein which means to loosen or untie and was firstly used in 1923. It aims at finding weaknesses and vulnerabilities in the cryptographic algorithms. Since the cryptanalysis concepts are highly specialized and complex, we concentrate here only on some of the high level concepts behind cryptanalysis.

This chapter comes as an overview for cryptography in general. In Section 2.1, a brief overview for cryptography including the security requirements, foundations and basic services is presented. The services that security can offer are related to which primitive you implement which is clearly stated in section. Additionally, we explain more about the famous KERCKHOFFS' PRINCIPLE. Then, in Section 2.2, a comparison between symmetric and asymmetric cryptography is made. This section presents a toolkit for the readers that will help understand the basic differences between these two types of systems. In this section we list the basic cryptography primitives that are classified according to symmetric and asymmetric titles. When speaking about symmetric cryptography, we explain the block ciphers and its modes of operation, confusion and diffusion primitives. Then, we explain the difference between block and stream ciphers. After that, we explain more about hash functions that are widely used in cryptography and the restrictions they face. Later, we explain the Message Authentication codes and what is their objective. After that, the explanation of the authenticated ciphers is also presented. Then, in terms of asymmetric cryptography, we list the public key encryption basic ideas, public key in-

frastructure and digital signatures. In fact, we explain why digital signatures are used and why they are so important. Also, a comparison between the symmetric and asymmetric ciphers' advantages and disadvantages is presented and we reach a conclusion that hybrid systems are more trusted than each of these systems when implemented alone. Finally, Section 2.3 concludes the Chapter.

2.1/ CRYPTOGRAPHY: FOUNDATION AND BASIC CONCEPTS

Historically, cryptography was considered as an art rather than a science. At the beginning, the secret forms were credited to ancient Egypt about 4000 years B.C [46, 72]. Scribes at that time were communicating by writing messages with the aid of hieroglyphs. This art of writing with hieroglyphs was inherited from father to son and was broken later on by Champollion [138]. Later, the Roman emperor Julius Caesar used another technique, the Caesar cipher, to encrypt his private data. It replaced every letter of a message by the letter which comes three positions later in the alphabet. The initial message can be recovered by performing the inverse transformation which today we call decryption. The Caesar cipher is marked as the first recorded use of a substitution cipher [84]. Modern cryptography was born in the first half of the 20th century [311]. One of the most famous examples from that time is the Enigma machine which was invented by the German engineer Arthur Scherbius at the end of World War I and patented in 1928 [2].

The real need of security just begun when small computers and inexpensive ones became available. They could be purchased by people for home usage from the 1970s, after a large-scale integration made it possible to construct an adequately powerful microprocessor on a single semi-conductor chip. Then, the rise of the Internet in the early 1990s set the range for a totally new era. Although during the first decades of modern cryptography it was basically for military applications, in the last decades, cryptography is used now in each domain (commerce, banking, industry, health care..) and is considered an inevitable component in our era. The history of cryptology is a fascinating story in itself, but not the scope of this thesis. A good reading on the history of cryptology is the monograph of Kahn [46]. Nowadays, cryptography plays a crucial role in every device we hold and we deal with, it is the heart of the computer and communication security. Undoubtedly, the growth of the Internet of Things augmented even more the need and the importance of cryptography.

As said earlier, cryptography investigates the theoretical techniques to guarantee a safe and secure communication between legitimate parties that are known as the *Transmitter* and the *Receiver*. Ensuring that the original messages appear unreadable and ambiguous for any other user is the sole aim of cryptography. However, cryptanalysis tries to retrieve or intercept the original message from this hidden form. Cryptography was first founded by Shannon [4] in 1984. He defined the basic cryptographic model which is still used today, which is mainly composed of two parties, 1) Alice and 2) Bob where Alice is the Encrypter and Bob is the Decrypter. However, when Alice tries to send a message to Bob secretly, a third party named Eve tried to eavesdrop this message m. Any third party can appear in an insecure channel and only the strength of the security will guarantee a safe communication between Alice and Bob exclusively. In fact, the main component here is the key z which is exchanged only between these parties. The objective here is

to ensure the authenticity of the message and prevent any adversary from exploiting this private transmitted data. Shannon model is represented in Figure 2.1.

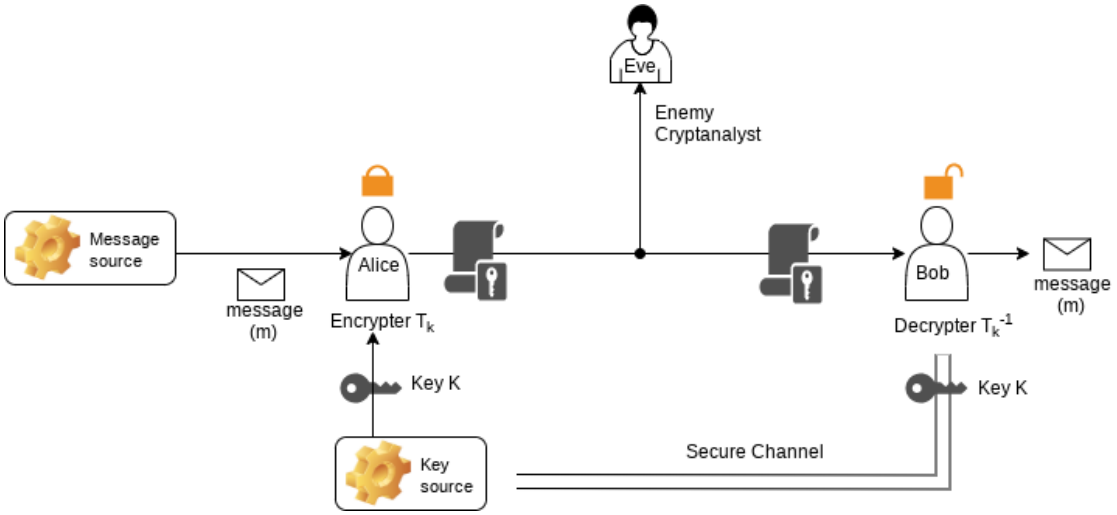


Figure 2.1: The traditional Shannon Model.

2.1.1/ KERCKHOFFS' PRINCIPLE

Today's cryptotography is titled under "Modern Cryptography", and one of its basic principles was proposed by the Dutch cryptographer Auguste Kerckhoffs [1]. Kerckhoff gave a practical, experience-based approach, including six design principles for military ciphers. The second principle he stated is now known to be Kerckhoffs' Principle. The principle states that:

"A cryptographic system should be secure even if everything about the system, except the key, is public knowledge."

This literally means that the whole security of any cryptographic system must depend only the secrecy of the key, and not the secrecy of the algorithm. There exist different arguments because of Kerckhoff principles [218, 311]. In fact, the algorithms are implemented in hardware or software, thus, making them vulnerable to reverse-engineering attacks. Firstly, keeping the secrecy of the whole algorithm is more difficult than maintaining the secrecy of a simple key. Secondly, when the key is leaked or exposed, it is more straightforward to substitute the key, than changing the whole algorithm. Thirdly, utilizing different keys and using the same algorithm among the communicating parties is much more logical than using a different algorithm at every end. Moreover, using a published cryptographic system/algorithm is safer than using a self-designed one. When the cipher is published, it will be under extensive security tests, thus, any flaw in it will be exposed. Finding mistakes in ciphers is easier than building a strong and robust algorithm.

However, the fact that any published algorithm is considered safe, is not correct. Many published algorithms are used and yet, they are attacked. This can be the case of insufficient amount of studies to prove the unreliability of this cipher. Or, simply, it was

not studied at all. Choosing what cryptographic algorithm to use is one of the hardest challenges. For example, the PC1 stream cipher, which was published on 1997 and was broken in 2012. It was already adopted in MOBI e-book format that was supported by Amazon Kindle and by the free software MobiPocket [267]. Another example is the ROT13, which is a mono-alphabetic substitution cipher. This cipher was used by the eBook vendor New Paradigm Research Group for ciphering their documents at least until 2001. This was revealed at a hacking conference [78], and the surprising part is that Windows XP used the same cipher on some of its registry keys as found in [137]. Some of the famous examples that were leaked or were reverse-engineered, then attacked are RC4 [228], DST [114], KeeLoq [43, 168, 173, 264], and Megamos [340]. Noting that we state these algorithms since these are examples of trusted ciphers that were later on attacked. RC4 was used by WEP in 1997 and WPA in 2003/2004 for wireless cards, and SSL in 1995 and its successor TLS in 1999. In 2015 it was prohibited for all versions of TLS by RFC 7465, because of the RC4 attacks weakening or breaking RC4 used in SSL/TLS. The last three algorithms were mentioned since they were used for car immobilizer transponders. More details about these algorithms can be found in [356]. Another example that can justify Kerchkoffs law is Telegram. It uses its own crypto-system MTProto which made cryptographic experts express doubts and criticism in its system, assuring that using self-brewed and unproven cryptography can put the whole system in jeopardy [330, 339, 346, 349]. Telegram has been attacked quite few times as stated in [363]. In 2015, there was a Man in the Middle attack against Telegram and still it suffers from several vulnerabilities.

2.1.2/ SECURITY SERVICES

Cryptography provides information security by using a set of techniques. A security service is a specific security goal that can be reached by employing cryptography. The primary four information security services that are targeted when using cryptography are: (1) Confidentiality, (2) Data Integrity, (3) Authentication, and (4) Non-repudiation [47]. These four basic concepts are the framework to derive the rest of the other security services as access control, anonymity, digital signatures, etc... [47]. An illustration for these security services is presented in Figure 2.2 with some of the cryptographic goals and primitives.

- Confidentiality: This security service guarantees that the content transmitted/shared is only accessed by legitimate users. Thus, it prevents any unauthorized access to this data. It is also referred to as secrecy.
- Integrity: This security service certifies by specific means that the data has not been manipulated or changed by any unauthorized party after the authorized user created, stored or sent it. Usually data manipulation means data insertion, deletion or replacement of the original data.
- Authentication: Authentication security service is related to identification and it can be divided into two classes: data origin authentication and entity authentication. Data origin authentication insures that an entity is the original source of a message. It implicitly provides data integrity, and referred to as message authentication. While, entity authentication aims at validating to one entity that the other entity in which it is interacting with owns a valid identity. Usually, entity authentication implies data origin authentication.

- Non-repudiation: It is the security service that prevents an entity from denying a previous action or commitment. This service is valuable in case of disputes or disagreements. Usually, in such cases, a third trusted party can validate the incident with a valid evidence it provides.

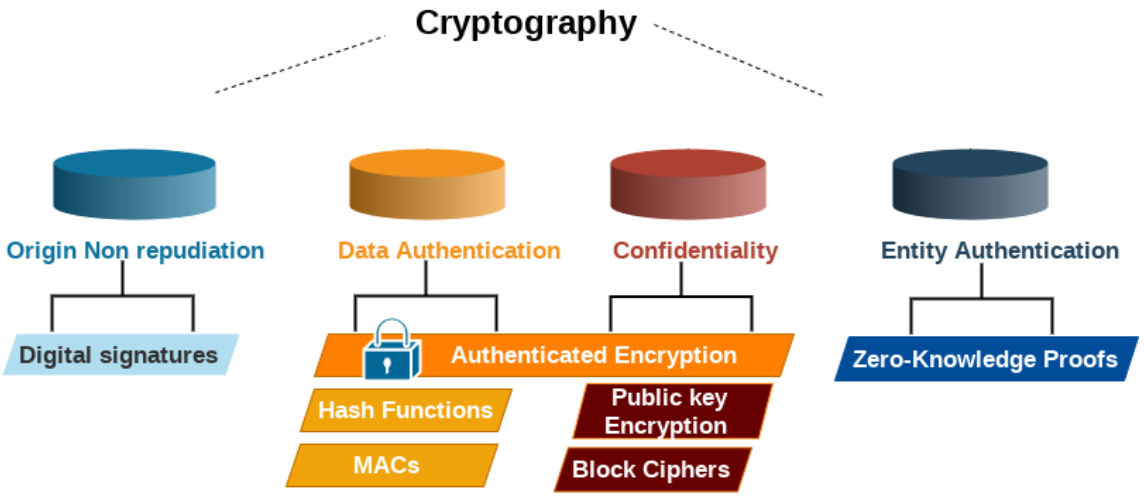


Figure 2.2: The four cryptographic services and some security primitives.

2.2/ CRYPTOGRAPHIC TOOLKIT

Cryptography can be divided into two main titles: (1) Symmetric Cryptography and (2) Asymmetric Cryptography. Mainly, these two branches differ by the operation of the cryptographic key. At the time symmetric cryptography uses the same key, or closely related keys, on both sender and receiver, the asymmetric algorithms do not. In fact, asymmetric cryptography allows a different key to be used on each side (encrypter, decrypter) [276]. However, the symmetric algorithms allow the encryption without the use of the key at all, like the case of hash functions. Both types of encryption have both positive and negative arguments, but, mostly in all systems, cryptographers aim to use both of these branches to exploit the advantages of these algorithms. In this section, a brief comparison between symmetric and asymmetric cryptography is presented.

2.2.1/ SYMMETRIC CIPHERS

Symmetric cryptography is one of the oldest forms of encryption. It is also referred to as secret key cryptography since the same key is used in both encryption and decryption processes. The algorithms that can be related to this class of cryptography are five: (a) block ciphers, (b) stream ciphers, (c) hash functions, (d) message authentication codes and (e) authenticated ciphers. The main argument in this kind of cryptography is the way of the transmission of the key used in the process of encryption/decryption.

2.2.1.1/ BLOCK CIPHERS

Block ciphers are the hub of the secret key cryptography. In fact, a block cipher divides the input (plaintext) into blocks of bits/bytes of the same length, before encrypting each block by using the secret key. The decryption must be applicable, therefore, the output must be invertible using the same secret key. Consider a block cipher with a block size n taking a plaintext (in bits) represented as $(m_0, m_1..., m_{n-1})$ and the outputted bits are considered as $(c_0, c_1, ..., c_{n-1})$. We can define the block cipher as the following:

Definition 1: Block Cipher

A block cipher with n bits block size and a key of size k bits, is an invertible mapping $F : \mathcal{M}\times\mathcal{K} \rightarrow C$ where $\mathcal{M} \in \mathbb{F}_2^n$ is the message space, and $\mathcal{K} \in \mathbb{F}_2^k$ and $C \in \mathbb{F}_2^n$.

It started in 1977 when the symmetric key cipher "Lucifer" was elected by NIST as the Data Encryption Standard (DES) [66]. Then, after many critics and attacks that appeared in many research works [26, 55, 130], and after double DES was also attacked [13], Triple DES was proposed as the prime encryption mode to be used in industries. Then, the requirement for more longer plaintexts and longer keys led to the famous Advanced Encryption Standard-AES originally called Rijndael [292, 391]. There are too many block ciphers to list them all, but DES and AES are the two most famous examples.

However, a block cipher is a family of $n - bit$ permutations of size 2^k where each secret key results in one permutation out of the total $2^n!$ n-bits permutations [263]. This means, that to get a purely random permutation from the whole set of n-bit permutations, the key used must be of size $log_2(2^n!) \approx (n-1)2^n$ bits which is considered a very huge number. It is not logical to have this amount of bits in a key, therefore, and to get closer to the ideal cipher, the goal is to draw 2^k permutations uniformly at random from the set of all $n - bit$ permutations.
A modern block ciphers iterates for several times. Every time it iterates, it means that a round function is repeated according to a pre-specified number. This round function consists of linear and non-linear layers that transforms the plaintext into the ciphertext using a round subkey. The construction types of block ciphers are divided into five categories:

1. Substitution Permutation Networks (SPN): This type of construction operates on the whole state. Two layers are used: Substitution and Permutation. The substitution layer is a non-linear layer that removes any linear relation in the plain text. It usually consists of an Sbox which is mainly made of boolean functions, that substitutes a small vector of the input bits (mostly $\leq$) with another vector values according to the used Sbox. The permutation layer applies a linear operation, which can be simple (bit-wise permutation) or complex (matrix multiplication), to shuffle/permute the state of bits.

2. Feistel Network (FN): The Feistel Cipher performs a diffusion operation on half of the data of each block, which results in a smaller round function. In fact, the block is divided into two equal halves (L_i, R_i) where i is the number of round. Then, the round function is applied only on one of the two halves. Consider R_i the one subjected to the round function using a subkey k_i, then, the output is xored with the other unchanged half L_i. Then, the two halves are swapped. This can be expressed as: $L_{i+1} = R_i$ and $R_{i+1} = L_i \oplus F(R_i, K_i)$.

3. Add-Rotate-XOR (ARX): This kind of cipher uses the addition, rotation and XORs operations avoiding the usage of Sboxes. They are known to produce a compact and fast implementations especially in terms of hardware efficiency. Today, ARX gains a lot of attention, since they are suitable for the variety of small devices that are available in the different networks.
4. NLFSR-Based: These are ciphers that utilize the building blocks of stream ciphers. They are based on nonlinear-feedback shift registers. They are efficient in terms of hardware implementations and the security of their inner components is based on stream cipher analysis.
5. Hybrid: This one combines different types of the aforementioned types of block ciphers. It usually aims at enhancing specific parameters in the cipher, such as throughput, number of rounds, execution time. etc...

⁕ **Modes of Block Encryption:**

Moving to the modes that a block cipher can operate in, choosing what type of mode to use is critical for block ciphers. The mode used can affect the speed of encryption/decryption, the robustness of the cipher against different adversaries, as well as the propagation of possible errors. The most famous modes that are standardized by NIST [76] are schematically described in Figures 2.3 to Figures 2.7. These modes are the Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher FeedBack (CFB), Output FeedBack (OFB), and CounTeR mode (CTR). In the following figures, encryption and decryption schemes are shown where ENC in these figures stands for Encryption and the same goes for DEC standing for Decryption. The key is generated by a pre-defined function and we assume that the plaintext is already divided into blocks $(P_0, P_1, P_2, ..., P_n)$ and the ciphertext produced is denoted as $(C_0, C_1, C_2, ..., C_n)$. Each of these modes of operations has advantages and disadvantages. However, the most known one is the drawback in the ECB mode where the created ciphertext is not completely blurred and identical plaintexts result in identical ciphertexts. Therefore, in other modes, we see that a parameter IV is used. To prevent similar patterns and eliminate repetitions, IV is used as an initial step in the message encryption. IV in CBC and CFB should not be predictable, in OFB it should be unique and for CTR, traditional IV is replaced by a $Nonce$ and a $counter$.

⁕ ***Confusion and Diffusion:***

The question that might be asked is **How do cryptographers know that the required level of security is reached?** Actually, Shannon [4] was the first one to answer this question and to formalize the ideas of Confusion and Diffusion which will define if a cipher meets the requirements of security. In fact, most of the block ciphers today are product ciphers, which means they are a set of subsequent operations for diffusion and confusion. Usually, the non-linear layer operation in a cipher attains the confusion property, which is mostly the substitution layer. As for the linear layer, represented mostly as permutation or ”mixing”, it usually attains the diffusion property. However, it is not so easy to distinguish between the components of the cipher that attains either confusion or diffusion alone. Using confusion and diffusion techniques can increase the power consumption and the execution time of the cipher. Confusion usually aims at making the relation between the ciphertext and the key complex as much as possible. For example, a small change in the input to an Sbox leads to a complex change in the output. In fact,

to have a robust crypto-system, a complex relation must be available between each bit/byte of the cipher text with the encryption key. In this way, a cryptanalyst will not tremble the security of this system. However, to spread this change to the entire state, a diffusion layer must be added. Usually, the easiest way of applying diffusion is by using a bit permutation layer, that can be realized easily on the hardware level. In this case, the adversary will find it very difficult to attack the system and it would take him/her plenty of time, since the redundancy has been diffused among a huge number of bit/bytes of the ciphertext.

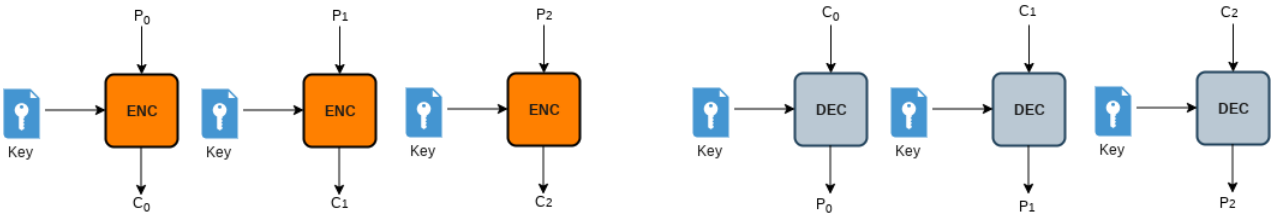


Figure 2.3: The ECB mode of operation - ENCRYPT and DECRYPT algorithms.

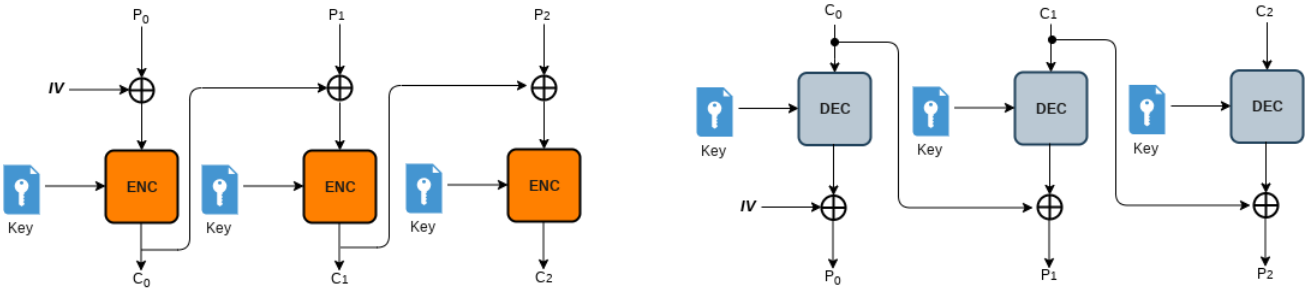


Figure 2.4: The CBC mode of operation - ENCRYPT and DECRYPT algorithms.

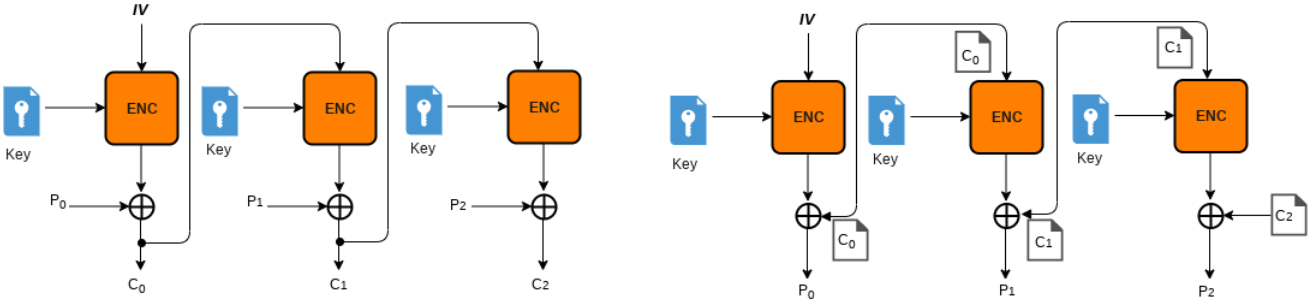


Figure 2.5: The CFB mode of operation - ENCRYPT and DECRYPT algorithms.

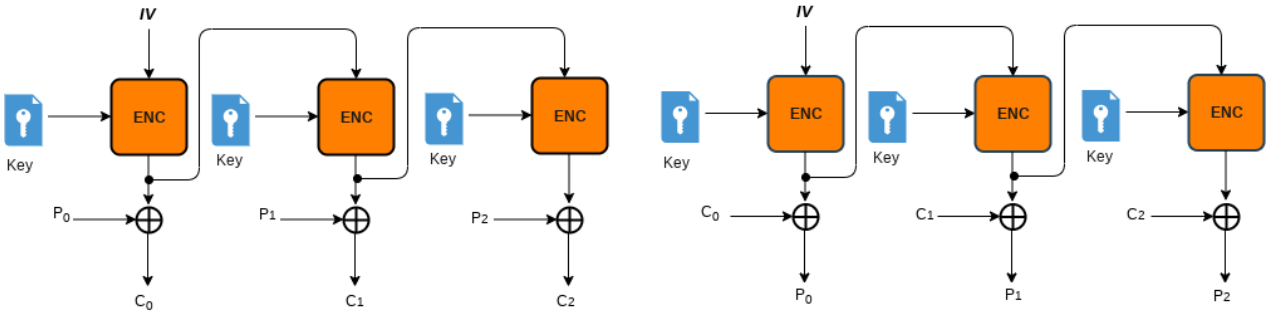


Figure 2.6: The OFB mode of operation - ENCRYPT and DECRYPT algorithms.

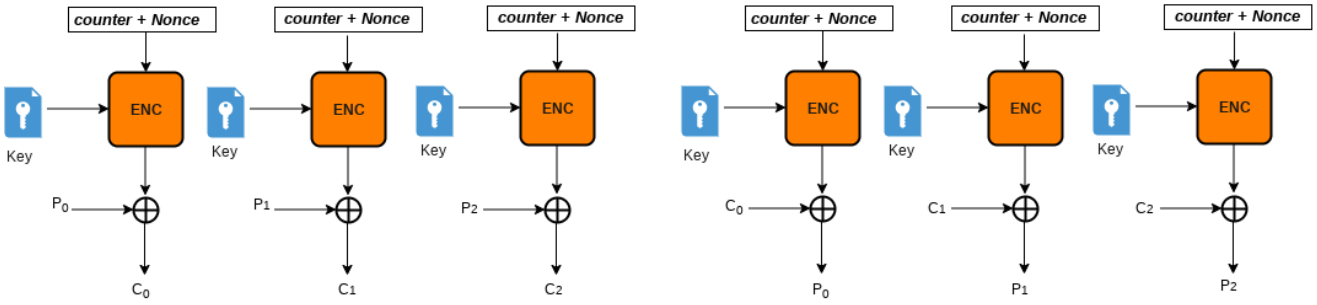


Figure 2.7: The CTR mode of operation - ENCRYPT and DECRYPT algorithms.

2.2.1.2/ STREAM CIPHERS

Stream ciphers were inspired by the One-Time Pad ciphers (OTP), that use a different key for every plaintext. In fact, OTP is the only provable secure cipher, since it uses the key only once, and the key has the same size of the plaintext being encrypted. This cipher only xors the key used with the plaintext bits. OTP provides a perfect secrecy if the keys used were fully random [4]. However, there are two main problems here, truly random numbers are so difficult to obtain, and the second one is the unpractical solution of sharing very large keys if the plaintext was very large. Therefore, stream ciphers came holding the solution to overcome these problems. They generate a pseudo-random stream of bits from a short secret key that can be easily shared between the parties. Modern stream ciphers use this key and mix it with a random initial value so that a very long pseudo-random key stream sequence (called the initialization phase) is generated. This sequence will be then xored with the given plaintext to produce the ciphertext.
There exist two kinds of stream ciphers: Synchronous and self-synchronizing stream ciphers. A self synchronizing stream cipher generates a key stream depending on the previous ciphertext, while the synchronous one generates the key stream independently of the plaintext or the ciphertext. Trivium [126] is a synchronous stream cipher selected as part of the portfolio for low area hardware ciphers by the eSTREAM project and was standardized by ISO/IEC [369].
Stream ciphers are usually built from Linear Feedback Registers (LFSR) and Non-Linear Feedback Registers (NFSR). LFSRs build the long sequences from the short one. Consider an LFSR having a length n, it then consists of n stages. It will be associated to a connection polynomial $c_n X^n + c_{n-1} X^{n-1} + ... + c_1 X + c_0$ where $c_i \in \mathbb{F}_2$ is used to update its state. Every stage will store one bit or a word with one input and one output. The clock will be controlling the flow of the bits. At every click, the data of stage 0 will be

the output which is a part of the full output sequence. Then, stage i will shift to be $i-1$ stage $\forall\ 1 \leq i \leq n-1$. Then, stage $n-1$ will be occupied by the feed-backed bit or word, s_j, that itself will be formed by xoring a fixed subset of the previous stages $(0, 1, ..., n-1)$ depending on the c_i's (the coefficients of the used polynomial). This can be expressed as: $s_j = \oplus_{i=1}^{n} c_i, s_{j-i}$. The maximum length of the produced sequence for an LFSR of length n is $2^n - 1$ iff the polynomial used is primitive. However, it is important to state that LFSR are vulnerable to the powerful attack $O(n^2)$ known as Berlekamp-Massey attack. It requires only 2n consecutive sequence bits (or words) to deduce the c_i's [79]. Thus, LFSR are used with a non-linear Boolean function to avoid this attack. Moving to NFSR, they are similar to LFSR but the feedback function is a non-linear Boolean function of the state.

2.2.1.3/ HASH FUNCTIONS

Mainly hash functions aim at providing authentication. However, until 1970s confidentiality and authentication were considered intrinsically connected. The first ones who pointed to hash functions functionality as a digital signature were Diffie and Hellman. However, the ones who provided definitions, analysis and constructions of cryptographic hash functions during 1970s were Rabin [10], Yuval [11], and Merkle [9]. In specific, Rabin proposed a hash function based on DES, Yuval took the analysis part and showed that the birthday paradox could find collisions in the hash function, while Merkle proposed the basic definitions that are used today (collision resistance, pre-image resistance, and second pre-image resistance). A hash function are either keyed or un-keyed hash functions (i.e. MACs). In both cases, its aim is to map a binary message of any arbitrary length to a small binary message of a fixed length, called hash value or message digest (that is why they are considered as compression functions). This hash value is considered as a unique fingerprint of the actual lengthy message. However, since the input size of the hash function is larger than the output size, there can be many messages that will have the same message digest. A hash function H aims to guarantee a number of cryptographic properties, to validate its information security. H must verify at least the following two implementation properties [109] :

1. Compression: H maps an input message M of arbitrary finite bit-length to a hash value h of fixed u bit-length.
2. Ease of computation: Given H and an input message M, H(M) must be easy to compute.

Notwithstanding, that also two important requirements are needed to realize a successful cryptographic hash functions: (1) the hardness to find collisions and the (2) appearance of randomness. Another three properties must also be proven within the proposed hash function:

- Preimage resistance (one way): Given a hash value, it is impossible to generate its actual message. In other words, reversing the hash function should be inapplicable.
- Second Preimage resistance (weak collision resistance): Given an input and its digest, it is hard to find a different input with the same digest.
- Collision resistance (strong collision resistance): It is unfeasible to find two messages that have the same hash value.

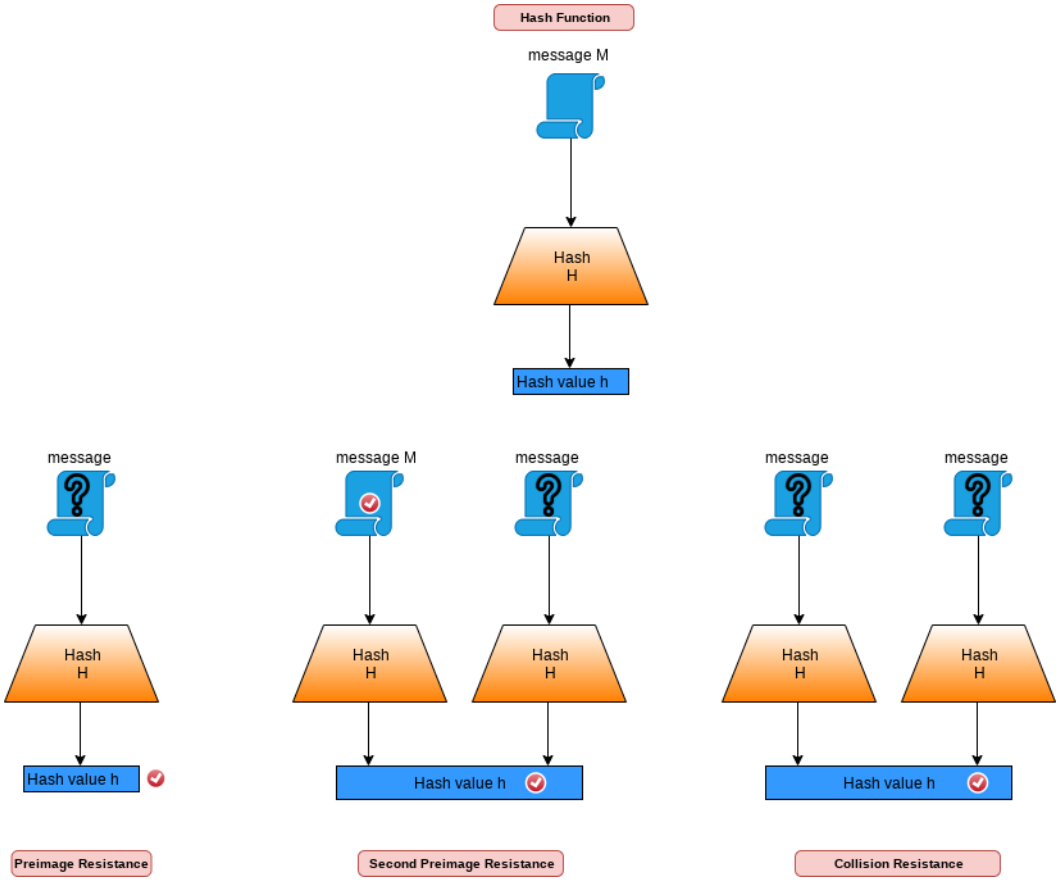


Figure 2.8: Security properties of hash functions.

Cryptographic hash functions are used widely in many security applications to ensure integrity and non-repudiation. Two well known hash functions are the Secure Hash Algorithm 2 (SHA-2) [255] and the Secure Hash Algorithm 3 (SHA-3) [332]. They are members of the Secure Hash Algorithm family of standards which was released by NIST. SHA-2 was proposed by the United States National Security Agency (NSA), and SHA-3 is a part of a bigger cryptographic primitive family named Keccak [297]. Also, the MD family (like MD2 and MD5) [28] is commonly used, as well as the RIPEMD-60 [44], HAVAL [30] and Whirlpool [135].

Birthday Problem This birthday problem (paradox) [100] is a familiar problem especially when we talk about hash functions. It refers to the probability of two people from n randomly chosen people, in a birthday party, having the exact birthday. This was translated to become a collision example in hash functions, the birthday problem is the basis for birthday attacks against secure hash functions. Given the number of a year with x days, the generalized birthday problem assume the minimum number of $y(x)$ such that in a set of randomly picked persons, the probability of having the same birthday is at least $\frac{1}{2}$. It can be said that $y(x)$ is the minimal integer y such that:

Definition 2: Birthday Problem

$$1 - \prod_{i=1}^{y-1} \left(1 - \frac{i}{x}\right) > \frac{1}{2} \tag{2.1}$$

The most common example that is widely used is when $x = 365$, a year by convention, and when $y = 23$, considering these are the people in the party, thus it yields to a probability of at least $\frac{1}{2}$. Birthday attacks usually take advantage of this argument, and perform a time-memory collision-search trade-off (i.e. saving memory at the cost of cryptanalysis time).

2.2.1.4/ Message Authentication Codes

Some protocols use a hash function to build a message authentication code (MAC) which can be also called a keyed hash function. They take a key and an arbitrary long message to produce a fixed size tag that will provide message authentication. It is used to authenticate a message, in other words, to confirm that the message came from the declared sender (its authenticity) and has not been changed. The MAC must have the forgery resistance property. This property guarantees that it is computationally impossible for an attacker to find a message and a tag pair without knowing the secret key used. One of the most known mechanisms used is the HMAC [51, 52, 81]. However, when using a MAC, several issues arise like how to choose the suitable key length and understand if the keys will be altered, which is usually something approximately impossible in most of the tagged-enabled applications. Also, there is a chance of having side-channel attacks on MACs which is beyond the burden of then thesis. In Figure 2.9 MAC function is illustrated to satisfy the confidentiality (by using encryption) and the integrity (by MACs) in the cipher system. It is important to note that to avoid replay attacks, there must be something within the data itself, that assures that the message is only sent once (e.g. time stamp, sequence number or use of a one-time MAC). Otherwise, although MACs can prove the authenticity of the message, but it would be vulnerable to replay attacks.

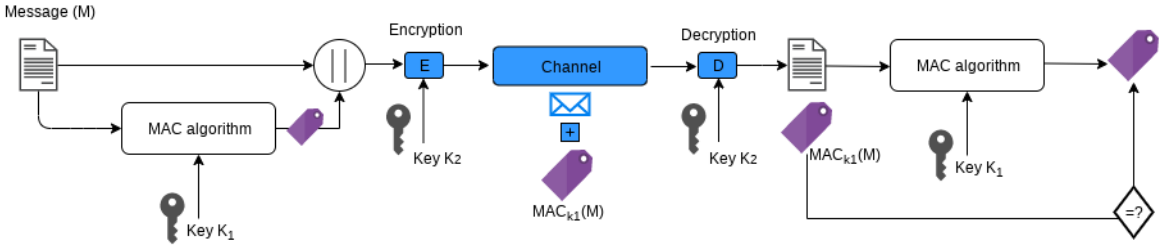


Figure 2.9: Message Authentication Code with Encryption.

2.2.1.5/ Authenticated Ciphers

Authenticated Encryption is a class of symmetric cryptographic schemes, that simultaneously provides message confidentiality, integrity as well as message authenticity. It is a fundamental component of almost every cryptographic protocol that is used in practice today. It is usually called authenticated encryption (AE) or authenticated encryption with

associated data (AEAD). Historically, the absence of authenticated encryption has been a problem for crypto-systems. Several proposals claimed that they met this goal and they were faced by absolute failure. For example, the first proposals of IPsec protocol [54] suggested that using CBC mode will guarantee the confidentiality and the integrity at the same time. This fallacy was due to the lack of formal security definitions where message authenticity and integrity were comprehended to be different [41]. Another example that can be given here is the WEP crypto-system in IEEE 802.11 standard, where a cyclic redundancy checksum and a stream cipher were combined in the aim of providing authenticated encryption. However, in these two examples, it was shown that they are inappropriate for having message authentication and message privacy [42, 144, 75]. Then, the authenticated cipher was motivated by the fact that combining a confidentiality mode with an authentication mode in a secure manner can render the system prone to errors and at the same time it will be a difficult task. The best way to reach an authenticated encryption scheme is by combining a message authentication scheme with a symmetric encryption scheme (i.e. *generic composition*). In fact, there are three natural ways to do so, and all the three approaches have been proposed in the practical crypto-systems and are listed below:

1. Authenticate-then-Encrypt: (AtE) The sender first computes a tag on the plaintext, then, this tag is appended to the plaintext. The result (the plaintext and its tag) is then encrypted. After the receiver receives the message, he/she decrypts it and gets the ciphertext, thus recovers the plaintext and the tag. If the tag is verified correctly, it returns the plaintext, otherwise it returns $\perp$ (a special ”invalidity” symbol) [77].

2. Encrypt-then-Authenticate: (EtA) In this case, the plaintext is first encrypted by the sender, then, the tag is computed according to the ciphered text. The tag is appended to the ciphertext and then sent. The receiver then receives the tag and the ciphertext. First, he/she verifies the tag, if correct, then he decrypts the ciphertext and returns the plaintext otherwise he/she returns $\perp$ (a special ”invalidity” symbol) [77].

3. Encrypt-and-Authenticate: (E&A) The sender computes the tag on the plaintext and at the same time encrypts the plaintext. The tag is appended to the ciphertext. The receiver recovers both the tag and the ciphertext, and decrypts the ciphertext. If the tag on the resulting plaintext is verified, the plaintext is returned, otherwise it returns $\perp$ (a special ”invalidity” symbol) [77].

2.2.2/ Asymmetric Encryption Algorithms

Asymmetric cryptography is also called public key cryptography and it usually studies systems using a pair of keys that are used to encrypt/decrypt data. This pair consists of a private key and a public key. The main issue in a public key cryptography system is the generation of this pair of keys in a way that it is computationally unfeasible to calculate the private key from the public one. The private key must be kept secret, no one should have it except the sender or the receiver, while the public key is distributed to other entities. A public key can be defined as a function that maps the plaintext to a ciphertext which can be done by anybody having the public key, but only the one with the private key can do the inverse. This makes the encryption often called a trapdoor since it can be

done in a one way direction. This roughness of reversing this operation is based on some mathematical problems that currently admit no efficient solution, or at least, there exist no algorithms that can solve them in a reasonable amount of time. Some of these problems are: Integer Factorization Problem (RSA [8]) and the Discrete Logarithm Problem (DLP) (ElGamal [16], digital signature algorithm-DSA [286]), or Discrete Logarithms on elliptic curves (elliptic curve cryptography- ECC [286]). There are three well known applications of asymmetric cryptography which will be explained briefly as they are out of the scope of this thesis. (1) public-key encryption, (2) digital signatures, and (3) public-key infrastructure (PKI). However, public-key cryptography is mainly used to grant the services of confidentiality, authentication, non-repudiation and secret key establishment.

2.2.2.1/ PUBLIC KEY ENCRYPTION

The public key cryptography was initially discovered by Ellis at the year of 1969, during his work at the British Government Communications Headquarters (GCHQ). In fact, RSA and Diffie-Hellman were uncovered by the GCHQ separately, several years before their development by the the cryptographic world [67], these were revealed just in 1997. Public key encryption is established by using two keys, a public one and another private. The public key will encrypt the message and this can be done by any one. However, the only one able to decrypt this encrypted message is the owner having the matching private key. Public key cryptography is known to be complex and computationally expensive, that's why they are often used to encrypt either small messages or a secret key. This secret key is then used by a symmetric cipher to encrypt/decrypt the messages in the corresponding crypto-system. It can be referred to as secure symmetric key transport-key wrapping. Assyemtric cryptography as indicated in by Diffie, it appeared in 1976 [19], then Merkle proposed another solution [12], which was known as "Merkle's puzzle". After that, in 1978, two major algorithms in the field of asymetric cryptography were proposed which are the RSA [8] and Merkle-Hellman [7]. As said earlier, RSA was based on the Integer Factorization Problem, while the Merkle-Hellman was based on a specific knapsack problem. Until now, RSA stood against different factors that make it insecure (length of the key changed by the modulo, padding, broadcast attacks..) while the other later, Merkle-Hellman was attacked later on and broke by Shamir [14]. Later on, elliptic curve versions of Diffie-Hellman's protocol were proposed. Finally, another DLP related cryptosystem to be mentioned is the ElGamal [16]. In Figure 2.10, a simple diagram shows the use of public and private keys in a public key encryption crypto-system.

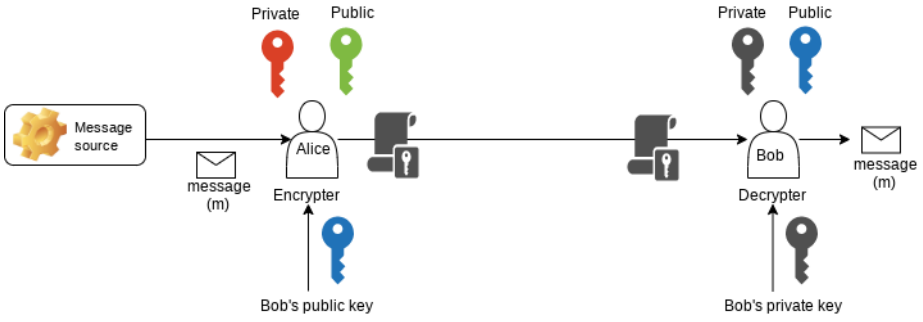


Figure 2.10: Public key encryption.

2.2.2.2/ DIGITAL SIGNATURE

Digital signatures can be considered as the public version of MACs. However, opposite to MACs, digital signatures can be verified publicly and non-repudiably. Being publicly verifiable serves the transfer of signatures and thus they are considered useful in many public-key infrastructures. The verifying and the non-repudiation makes the digital signatures very useful in cases such as contract signing. The digital signature can usually obtained by encrypting the plaintext using the private key of the sender. In most of the cases, this message will be hashed just before signing. Thus, anyone having the public key can verify that signature whether valid or not. Additionally to the sender's public key, the verification function takes a signature and a message. It validates whether the signature was generated from the the same message or not using the secret/private key. This verification function assures that the message was not altered or tampered by a third party. This can be summarized by three algorithms that are involved in digital signatures:

1. Key generation: This algorithm generates a private key along with its corresponding public key.
2. Signing: This algorithm generates a signature when receiving the private key and the message that is being signed.
3. Verification: This algorithm verifies the authenticity of the message by confirming it along with the signature and the public key.

The digital signature process is represented in Figure 2.11, and note that the plaintext is not encrypted, if the sender wants to encrypt the plaintext he/she has to use the public key of the receiver. If we want to state some of the most known digital signatures, first, we start by the Diffie and Helman's paper, where they proposed a method to allow the construction of signatures from encryption trapdoor permutations [5]. Then, based on this proposal, the first signature was proposed by Rivest, Shamir and Adelman, which is the famous RSA [8] in 1978. Derived from fair zero-knowledge identification, a new method was proposed to obtain signature schemes (by Goldwasser, Micali, and Rackoff) [20]. Then, in 1987, based on the hardness of extracting modular square roots, Fiat and Shamir built a zero-knowledge identification protocol in [17] which also contains a well-discussed digital signature scheme. In 1991, the first RSA based digital signature international standard showed up [97]. Until now, there have been many proposals that have additional properties. We quote a few like blind signatures [15], designated verifier signatures [45], ring signatures [80], group signatures [80], and automatic signatures [82].

2.2.2.3/ PUBLIC KEY INFRASTRUCTURE-PKI:

Kohnfelder was the very first to established the very important concepts of PKI in his thesis [6]. A PKI system allows the distribution and the identification of the public keys. In such systems, the parties are able to send messages securely as well as verify the identities of the remote parties. In a public key infrastructure system, there is a third party which makes sure that the owner of the key pair (public key and private key) is certified by a public-key certificate. The public-key certificate contains information about the public key, the owner's identity, and the validity period. This data will be signed by a Certification Authority (CA), which role is to issue, store or even revoke the public-key

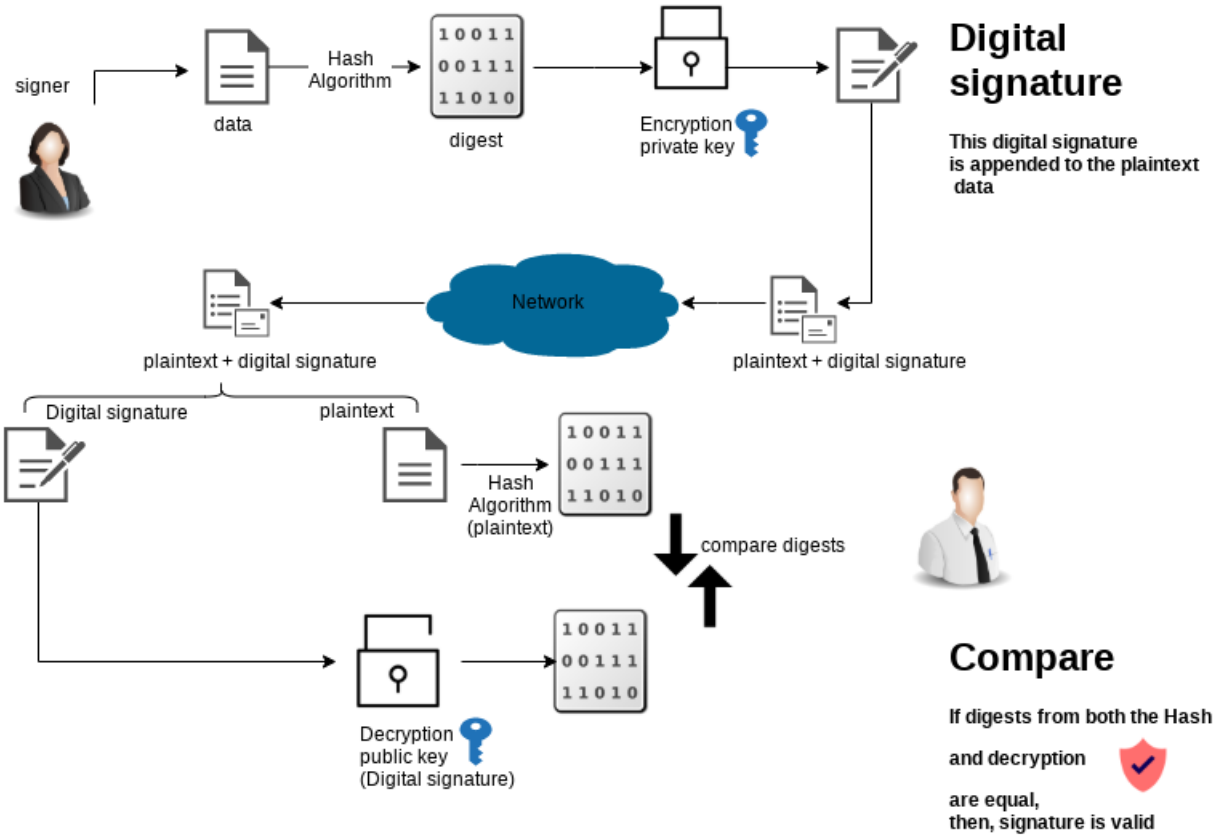


Figure 2.11: Process of digital signing and verification.

certifications. The parties involved in a PKI must own a public-key certificate that is issued from this CA. Additionally, every participant in this system must know the CA's public key, in order to verify the certificates for the other recipients. It is important to state that digital certificates are the heart of the PKIs. These certificates confirm the identity of the user, also it connects the user identity with his public key that is in his certificate. Mainly, these parties exist in a PKI system:

1. Trusted Party: (CA) It is the certificate authority and it acts as the root of trust.

2. Registration Authority (RA): It can be in some cases a subordinate trusted party, but certified surely by the CA. It can issue a lower level certificates, or in other systems it is just used to record the users and the public keys without signing the certificates.

3. Certificate Data Base: It stores the certificate requests, issue or revoke certificates.

4. Certificate Store: This one is located in the client's device (mainly the computer), which stores certificates, public keys and the private key of the user.

Cipher primitive	Authentication	Integrity	Non-repudiation	Confidentiality
MAC	$\oplus$ *	✓	✗	✗
Hash	$\oplus$ *	$\oplus$ *	✗	✗
Authenticated Cipher	✓	✓	✗	✓
Block ciphers	$\oplus$ *	$\oplus$ *	✗	✓
Stream Ciphers	$\oplus$ *	$\oplus$ *	✗	✓
Public key encryption	✓	$\oplus$ *	✗	✓
Digital Signature	✓	✓	✓	✗

Table 2.1: The security services provided by different cipher primitives, symmetric and asymmetric.

2.2.3/ A Comparison Between Symmetric and Asymmetric Cryptography

As seen, both symmetric and asymmetric cryptography have pros and cons. That is why in modern systems and in most of the used systems today, both are used at the same time. For example, we cannot ignore that symmetric encryption preserves confidentiality in a faster manner than the asymmetric ciphers. As said earlier, symmetric ciphers usually use smaller keys, but how to share this key is the risen argument and how to keep it secret between the entities especially when having a large number of involved parties. It is clear that public key cryptography is used with the sole aim of non-repudiation and to have a secure channel to exchange messages. Before, the rise of public key cryptography, users used to send and distribute sometimes the whole code book by secure means. This would be very risky when in a military or war state. Allowing the enemy to intercept the key (“red” keys in military parlance) would lead to your death after he deciphers all the exchanged messages, or simply you will be in a denial of service state when you avoid sending private information to other parties.
Today, the most commonly used method is combining both systems in a system called ”Hybrid Encryption”. The keys will be exchanged in an asymmetric algorithm to secure the channel and then this key is used to encrypt/decrypt using a symmetric algorithm. This is the basis for most of the connections, Internet communication and electronic transactions today. However, as cryptography evolves, quantum key distribution also has been proposed to exchange this secret key, which is a a technology that uses light in fiber optics, depending on the nature of photons.
In Table 2.1, an overview of the main security services when using symmetric or asymmetric encryption is presented. It is clear that when using a hybrid encryption, almost all the security services can be granted to the new crypto-system. Note that ✓ means that the cipher primitive can grant the security service all by it self, ✗ denies the ability of the cipher primitive to give the desired service, and $\oplus$ * stands for the ability to give this security service in case of combining the primitive with another security primitive or simply by changing the mode of operation.

2.3/ Conclusion

In this chapter, a cryptographic general idea has been presented. First, we presented the foundations and the general concepts of cryptography. The basic security services

are listed along with an explanation for each service. The main principle of cryptography Kerckhoffs' principle is explained. After that, we differentiated between the symmetric and the asymmetric algorithms and their ways of employment. In the symmetric cryptography we explained the block ciphers and their modes of operation. Then we explained the stream ciphers. Also, the message authentication codes and the hash functions are illustrated as well. Hash functions are also explained with the constraints they face. Each one of these cipher primitives can provide one or more security service. While talking about asymmetric cryptography, we explained the public key encryption, public key infrastructure and how does digital signature operate. We found that hybrid systems are the best chance to get a fully protected crypto-system. Symmetric algorithms are widely used today as well as asymmetric crypto-systems, and both are being deployed to face the new and dangerous attacks and to be able to have a secure communication. This kind of systems is very famous and is called a hybrid system. To sum up, this chapter comes to lay the foundations for the next chapters that manage other deeper aspects of cryptography.

3

LIGHTWEIGHT CIPHERS

"Light As A Feather, Stiff As A Board."-The Magician's Own Book

If you grew up in the 90s, then you have definitely heard of the game *"Light As A Feather, Stiff As A Board."* It was one of those slightly spooky, kind of anxiety-inducing games that was played at most of the parties. However, this can make a perfect description for this section or in fact for this thesis: Lightweight Cryptography. As new terms emerge, we need new solutions that can adapt to the demanding need for the change. On one face of the coin, we need very lightweight implementations that are "light as a feather", however, on the other side of the coin, we need the "stiff as a board", which cryptographically speaking, a high level of security has to be guaranteed. A trade off between these aspects must be made, to have a whole trusted and secure system. In this section we state most of the lightweight algorithms that have been proposed by different researchers in different domains.

Lightweight cryptography has emerged recently due to the massive need of new algorithms that can fit in today's devices. In fact, the embedded devices that are used widely in different platforms have set their own needs. The interconnection of these embedded devices leads to the famous vision of Mark Weiser's vision of ubiquitous computing (ubicomp) [68]. It is widely agreed that the ubiquitous computing is the upcoming paradigm in information technology. When we reach a day that 98.8% of all manufactured microprocessors are being employed in embedded applications and the remaining 1.2% is employed in computers, then, we can say that we are in a need for new solutions that can fit. Back to 2002, Ross Anderson foresaw that "your fridge, your heart monitor, your bathroom scales and your shoes might work together to monitor (and nag you about) your cardiovascular health" [88]. Today, we live in the era of the connection of everything we see, the term being using is the famous IoT-Internet of Things. RFID tags, wireless sensors, embedded sensors and devices, etc.. these are all devices with limitations in their processing power, life-time and memory constraints.

In this Chapter, we focus on listing the different lightweight cryptographic algorithms that are proposed up to the date this thesis is written. In Section 3.1, we describe the emerging term Internet of Things-IoT and we list the constraints that motivated many researchers to work in this field. Then, in Section 3.2, we list most of the lightweight algorithms that we have knowledge about. Then, in Section 3.3 we draw a summary for this chapter.

3.1/ INTERNET OF THINGS

Internet of Things [193] term has been a buzzword for a quite time now. The main objective behind this term is the ability of having an Internet connection among a huge number of devices. These devices are often referred to as Things, and the connection among them creates the ubiquitous systems. The sole aim of moving towards IoT is to obtain an easier life, that gives you a lot of comfortability in your daily routine life. Imagine a refrigerator for example giving you an alert for having less milk meanwhile [348]. These devices, gadgets, sensors... do not use the classical Internet to exchange or live-stream data among each other, in fact, there is no standard communication that targets all of them. A huge heterogeneous network will be constructed with different abilities, hardware, and protocols among its participants. The one common foundation that is used among them is the usage of the Internet layer (i.e. IP/IPv6) connectivity. These devices are clustered in small networks, but yet they also need to communicate among bigger networks. The communication is done across special hubs and gateways that will send the data across this heterogeneous IoT technologies and the Internet. IoT is a life changing concept that will affect not only the people, but the industries as well. All new demands rise with this interesting term and brainstorming technology. The term IoT is mainly used for devices that would not usually be expected to possess an internet connection, and they can communicate with the network independently of human action. In Figure 3.1, an architecture with the services that IoT can provide is presented.

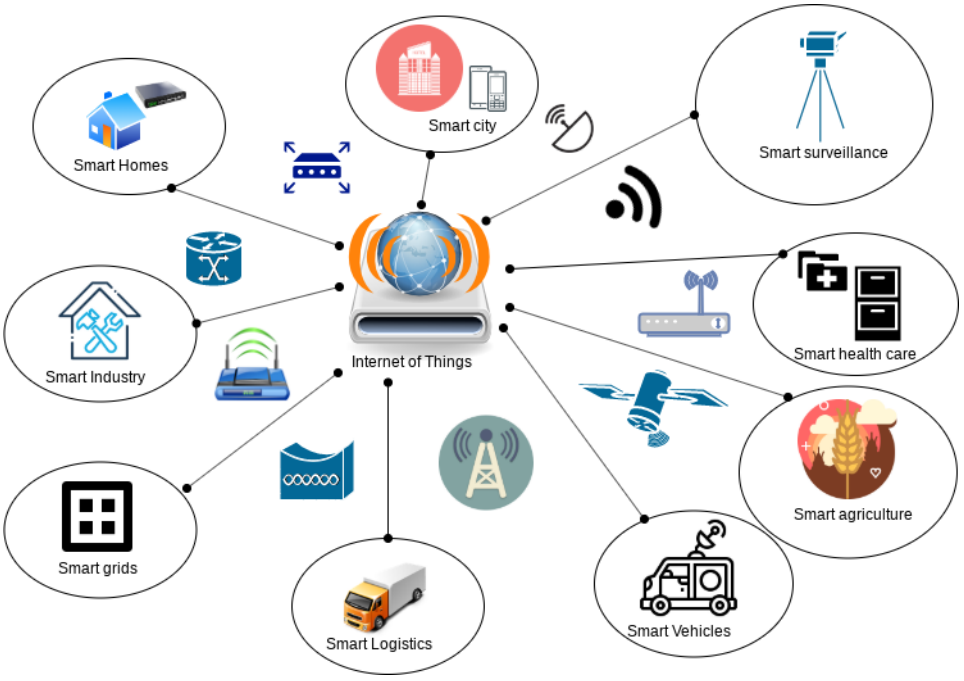


Figure 3.1: An architecture for the Internet of Things.

3.1.1/ Constraints and Motivations

When talking about IoT, the first idea to come to the mind is the amount of heterogeneous devices being used in such systems. These IoT devices are used for specific applications under many constraints. Limited amount of resources is one of them (i.e. energy), and many other combination of factors that are imposed to financial constraints at the end. However, the amount of resources that are dedicated to security are very low, in fact, they are just a fraction of the total available resources. Flaws because of the negligible cost for these devices have emerged. For example, home devices like refrigerators, ovens, dishwashers are open to hackers, and mainly webcams are easily hacked to be specific. Researchers have found that 100,000 webcam are easily vulnerable to hacking [361] additionally to the children smart watches that can track, locate, eavesdrop conversations as well [362]. All these sensitive devices put a great danger on its users, especially children that can be easily kidnapped in such cases. Therefore, the security, cryptography in specific, should provide the desired protection and they must face the stringent constraints without jeopardizing the security. These hardware constraints are mainly because of the hardware implementation, the silicon area in specific, latency, power consumption and limited memory. Concerning energy, many sensors use either batteries or generate their own energy (solar energy). For the software implementation, some constraints are the code size, the execution time, energy consumption and the number of rounds of the cipher. Here comes the problem of optimization that usually ends at last with having one goal achieved. For example, optimizing the code can jeopardize the security, less number of rounds can render the cipher vulnerable to attacks. It is very hard to optimize for more than one objective, when latency versus security and security versus speed. As a result, a trade-off has been made among all the cryptographic solutions proposed to meet with the hard constraints that are available in small limited devices [204, 245].

3.2/ State-of-the-Art

Lightweight cryptograhy has been proposed to face these constraints that normal and conventional ciphers cannot face especially in IoT. These algorithms must comply with the hardware/software restrictions and at the same time preserve a high level of security. In [119], Gligor defined ”lightweight cryptography as cryptographic primitives, schemes and protocols tailored to (extremely) constrained environments”. The main challenge is how to face the conflict between the different metrics from security and hardware available. At the same time, they must face the new and novel attacks that are growing each and everyday due to the massive deployment of super computers and GPUs. Below, we list most of the lightweight algorithms that are proposed today, but first we explain one of the most used cipher among the world which is the AES: Advanced Encryption Standard. Then, the lightweight ciphers are listed under **Block lightweight ciphers, Stream Lightweight ciphers, Dedicated Authenticated Encryption Schemes, Lightweight Hash Functions, and Lightweight Ciphers for Governments**. In the last part of this subsection, we state some differences between lightweight and ultra-lightweight encryption based on some assumptions.

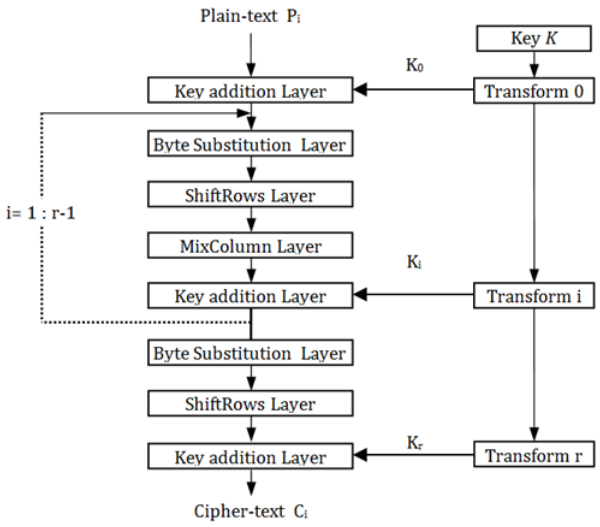


Figure 3.2: AES architecture.

3.2.1/ Advanced Encryption Standard

The first encryption scheme that was used extensively before AES was the Data Encryption Standard (DES) which was proposed for image cryptography [35]. Based on DES, researchers proposed other ciphers such as Triple DES [354] and the algorithm proposed by Qian Gong-canister et al. [94] who integrated DES with a chaotic map to enhance the quality of the image and increase the key space. However, DES was not efficient to treat large amount of data since it will be very expensive especially for modern applications [161]. Therefore, Vincent Rijmen and Joan Daemen, proposed the famous standard AES [292]. In Figure 3.2, the cipher is represented. AES is a symmetric block cipher, that takes blocks of 128,192,256 bits and have 10,12,14 rounds respectively of processing. Each round except the last one has four operations to undergo: **Byte Substitution layer (S-Box), Shift Row layer, Key Addition layer and MixColumn layer**. The MixColumn layer will be eliminated in the last round. First, the round key is Xored to the state, then a byte-to-byte substitution is performed using an 8 bit look-up table to attain the diffusion property. After that, a permutation is realized by the Shift Row layer at the byte level also. Lastly, a MixColumn layer is applied that combines blocks of four bytes by the aid of a matrix operation. The diffusion property is preserved by the ShiftRow and the MixColumn operations since they spread the change of the Substitution layer to the rest of the bytes. AES became as a federal government standard as of 2002 after the approval by the Secrecy of Commerce. It is also included in the ISO/IEC 18033-3 standard. It is also the first and only public available cipher approved by the National Security Agency (NSA) for the top secret information when used in an NSA approved cryptographic module.
Other algorithms were also proposed based on AES. In [161] an algorithm is proposed based on AES, based on a key stream generator named as (A5/1, W7) in order to improve the AES performance in securing images. In [250] Subramanyan et al., proposed an expansion key that is modified in order to enhance the avalanche effect and improve the encryption quality. Another proposed authetication scheme based on AES is ALE: AES-Based Lightweight Authenticated Encryption [307], which is an online single-pass authenticated encryption algorithm that supports optional associated data.

However, apart from being strong and efficient, AES is considered to be heavy on small devices since it requires the storage and the processing of heavy computations. Its computational time is considered to be high since it uses 10 rounds of iterations at least. Additionally, due to the spacial characteristics of the images where the most important part is the content and not the pixel itself, AES is not the best choice to pick. This cannot fit in the real-time delivery and the stringent constraints that are available today [167].

3.2.2/ BLOCK LIGHTWEIGHT CIPHERS

Block ciphers have been used extensively to fulfill lightweight cryptography. Below, we list most of the block ciphers that were designed and implemented for the sake of limited devices.

1. **3-Way (1994) [31]:** Proposed by Joan Daemen, Rene Govaerts and Joos Vandewalle, they apply the cryptographic finite state machine approach as in DES to design a simpler approach, based on simplicity, uniformity, parallilism, distributed non-linearity and high diffusion. 3-Way is block cipher that takes 96 bits key and uses 3-bit non-linear Sbox and a linear mapping (modular polynomial multiplication). However, one of the analysis shows an attack against this cipher [50].

2. **RC5 (1995) [36]:** Proposed by Ronald L. Rivest, RC5 has been well known since 1995. RC5 uses data-dependent rotations and it is fully acclimatized in word size, number of rounds, and key length. This was a great advantage for RC5 among the rest of the ciphers that is also suitable for hardware and software implementations. Although RC5 has been able to be cracked when 64-bit key is used in RSA laboratories; Secret-Key Challenge after 1757 days, RC5 with 128 bit key has stood against the attacks. Then, RC5 was attacked in [48, 57, 60].

3. **Misty1 (1997) [53]**: Proposed by Mitsuru Matsui, MISTY1 has been recommended by the CRYPTREC in 2003. It is a 64 bit cipher and the predecessor of KASUMI, the the 3GPP-endorsed encryption algorithm [3GPP 1999]. It is designed for high speed implementations (both hardware and software) by using only logical operations and look-up tables. MISTY1 is an open standard documented in RFC2994 [74]. It operates with 8 rounds in CBC mode with 128 bits key. In [365, 70], writers showed that they were able to attack MISTY by an integral analysis attack and differential one.

4. **BKSQ (1998) [56]:** Proposed by Joan Daemen and Vincent Rijmen, based on SQUARE algorithm [49], BKSQ is a block cipher that has 96 bits key length, operates on 96,144,192 block size and has respectively 10,14,18 number of rounds. It can be also used as a (2nd) pre-image resistant one-way function.

5. **Khazad (2000) [71]:** Proposed by Paulo S.L.M. Barreto and Vincent Rijmen, The KHAZAD Legacy-Level Block Cipher is a 64-bit (legacy-level) block cipher that accepts a 128-bit key. Khazad has been submitted as a candidate cryptographic primitive for the NESSIE project [87]. It takes 128 bits key and 64 blocks size, then operates for 8 rounds. The basic strategy behind this cipher is building it upon the the Wide Trail strategy [39] which states that the round function has to have different invertible transformations. Each transformation must own a specific function

(i.e diffusion layer, non-linear layer, round key function). By applying the Wide Trail strategy, the components will be independently recognized.

6. **Iceberg (2004) [110]:** Proposed by Francois-Xavier Standaert, Gilles Piret, Gael Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat, Iceberg operates with a 128 bit key and 64-bit text block size and 16 rounds. It is considered to be an involutional block cipher for optimized hardware implementations. ICEBERG permits to change the key at every clock cycle without any performance loss and its round keys are derived ”on-the-fly” in encryption and decryption modes (no need to the storage of the round keys). In [281], a linear cryptanalysis was claimed against Iceberg.

7. **mCrypton (2005) [121]:** Proposed by Chae Hoon Lim and Tymur Korkishko, mCrypton [64] is based on Crypton but with simpler components. It is a 64-bit block cipher with three key size options: 64, 96, 128 bits. mCrypton was proposed specifically for the use in resource-constrained tiny devices (RFID tags and sensors) and it operates for 12 rounds. However, in [202] it was shown that 8-round mCrypton with 128-bit key is vulnerable to related-key rectangle attack.

8. **SEA (2006) [136]:** Proposed by Francois-Xavier Standaert, Gilles Piret, Neil Gershenfeld and Jean-Jacques Quisquater, SEA is a low cost encryption cipher used for small codes and memory and targets the processors with limited instruction set (AND, OR, XOR, rotation, and modular addition). SEA uses a 96 bits key, 96 bits of block size and 93 rounds. Note that, the number of rounds is quite a high number in our cryptographic point of view.

9. **CLEFIA (2007) [157]:** Proposed by Taizo Shirai, Kyoji Shibutani, ToruAkishita, Shiho Moriai, and Tetsu Iwata, CLEFIA has been proposed with similar criteria as AES. It supports the key lengths of 128, 192 and 256 bits, which are compatible with AES. It uses blocks of 128 bits and has 18, 22, 26 number of rounds. The fundamental structure of CLEFIA is a generalized Feistel structure consisting of 4 data lines, in which there are two 32-bit F-functions per one round. Note that diffusion is employed since it uses different diffusion matrices, and two different Sboxes. However, the diffusion process remains an essential and a difficult process to fulfill in an efficient manner [129].

10. **PRESENT (2007) [141]:** PRESENT was the first ultra-lightweight to be proposed and had so much attraction from researchers. Both security and hardware efficiency had been equally important during the design of the cipher and at 1570 GE, the hardware requirements for present were competitive with the leading stream ciphers. PRESENT uses 80, 128 key length with 64 bits block size and 31 round number. It was basically a SP-network. However, having 31 round on a small device that works on limited battery and memory, is not a good solution in terms of efficiency, and it is far from being ultra-lightweight compared to the recent works.

11. **PRINTCipher (2010) [224]:** It was proposed for the IC (integrated circuit) printing in specific. It uses 48, 96 bits as a key length, and for the block size it uses 80, 160 bits while running for 48 or 96 rounds respectively. However, it did not meet the desired echo by the researchers. In fact, the key length is really small (48) which makes it vulnerable to brute force attacks.

12. **KLEIN (2011) [242]:** KLEIN was proposed for resource-constrained devices such as wireless sensors and RFID tags. It adopts a key of 64, 80, 96 bit and a block size of 64 bits size, with a 48, 96 rounds. Without a single doubt, 96 rounds for a small device, even if it was containing the simplest operations, will definitely be expensive and time consuming. additionally, in [235], an attack for 8 rounds of Klein was deducted.

13. **LED (2011) [243]:** The LED cipher was designed for offering the smallest silicon footprint among comparable block ciphers. It uses 64, 128 bit key and a block size of 64 bits. The number of rounds is 32, 48. The components of LED are very similar to those of AES thus, it is hard to claim that this is an ultra-light weight cipher, when using Sboxes, ShiftRows, and MixColumns. Additionally, in [277] a differential crypt-analysis is presented against LED, and in [271] an algebraic fault attack was deducted on LED.

14. **PICCOLO (2011) [249]:** As an ultra-lightweight cipher, Piccolo was proposed for limited devices. It supports 64-bit block with 80 or 128-bit keys, and has an iterative structure which is a variant of a generalized Feistel network with a permutation based key schedule. It has a 25 or 31 number of rounds. Additionally, They prove in their work that Piccolo is resilient against different number of attacks (differential, linear, man-in-the-middle, related-key). Yet, in [303, 270] Biclique attack were able to succeed against Piccolo (they showed that slow and limited diffusion in the key-schedule and the encryption process in the targeted algorithm lead to relatively long bicliques with high dimension and an efficient matching check with pre-computations.)

15. **PRINCE (2012) [268]:** PRINCE was proposed to be a low-latency block cipher for pervasive computing applications. It uses 12 rounds, 128 bit key size and a block size of 64 bits. The main objective behind this proposal is an instantaneous encryption/decryption without a warm up state. A ciphertext is computed within a single clock cycle. However, in [304], PRINCE was under a differential fault attack. The attack uniquely determines the 128-bit key of the cipher using less than 7 fault injections averagely.

16. **LEA (2013) [296]:** LEA works on 128, 192, 256 bit keys with a block size 128 bit. The number of rounds is 24, 28, 32 respective to the key used. The authors' experiments show that LEA is faster than AES on Intel, AMD, ARM, and ColdFire platforms. LEA is an ARX cipher (uses modular Addition, bitwise Rotation, and bitwise XOR) that operates on 32-bit words. Those operations are well-supported and fast in many 32-bit and 64-bit platforms. However, there were different attacks against LEA. In [336] a side channel attack was held against the latter, and in [319], there was a differential fault analysis attack. Moreover, and still we do not consider this high number of rounds suitable for very limited devices.

17. **RECTANGLE (2015) [341]:** RECTANGLE is based on an SP-network. The substitution layer used consists of 16 4×4 Sboxes in parallel. For the permutation layer, it is composed of 3 rotations. The authors claim that RECTANGLE offers great performance in both hardware and software environment, which provides enough flexibility for different application scenarios. RECTANGLE uses 80, 128 bit key and 64 bits block size. Concerning the number of rounds, it is 25. Since this is an SP-network, the number of rounds is quite large. Usually, Feistel type of networks

requires a higher number of rounds, and this reduced the efficiency of the cipher. Having 28 round for an SP network, is considered very expensive in terms of hardware and software. Yet, in [322], a Differential Power Analysis (DPA) attack was induced which reduced the key search space from 2^{80} to 288 key which will be a very easy brute force attack.

18. **RoadRunneR (2015) [327]:** Based on the facts that some lightweight ciphers have low security margin (PRIDE and PRESENT), RoadRunneR was proposed by Adnan Baysal and Suhap Sahin. It targets low cost 8-bit processors with a 64-bit Feistel design with initial and final round whitening keys. It operates with 80, 128 bit keys and 64 bit block size with 10 or 12 rounds. Also, this cipher was attacked in [353] with truncated differential characteristics.

19. **SKINNY (2016) [343]:** It has been proposed as a competitor to Simon cipher. It belongs to a new family of tweakable block ciphers presented at CRYPTO 2016 designed under the TWEAKEY framework [321]. SKINNY has flexible block and key sizes. It operates with 64-384 bit key length and has 64,128 bit block size that are processed into 32-56 rounds. The designers' aim was to provide strong bounds for all versions, and not only in the single-key model, but also in the related-key or related-tweak model. It reached an extremely small area for serial implementations and a very good efficiency for software and micro-controllers implementations. In fact, SKINNY has the smallest total number of AND, OR, XOR gates used for encryption process. Later on, in [364] the researchers found a 16 related-tweakey impossible differentials of 12-round SKINNY, based on which they could attack 18-round SKINNY-64-128.

20. **SPARX (2017) [344]:** SPARX uses 128, 256 bit key length and block size of 64, 128 bits and iterates 24-40 rounds. It states that the wide trail design strategy (WTS), that is at the basis of many S-box based ciphers, including the AES, is not suitable for ARX designs due to the lack of S-boxes in the latter. They addressed the mentioned limitation by proposing the long trail design strategy (LTS); i.e. a dual of the WTS that is applicable (but not limited) to ARX. They proposed two different strategies to build ARX-based block ciphers with provable bounds on the maximum expected differential and linear probabilities. However, there were some attacks on reduced versions of SPARX. In [383], authors proposed a technique to perform Correlation Power Analysis (CPA) on the SPARX-64/128 cipher. They used a combination of first-order, second-order and modulo addition CPA methods. After all, they were able to extract 128 key bits of SPARX-64/128 cipher with low complexities in general; key guess complexity of 2^{12} and 65000 $\approx 2^{16}$ power traces.

3.2.3/ STREAM LIGHTWEIGHT CIPHERS

One of the most important competitions that occurred was the eSTREAM competition in 2008. Below we list the most famous lightweight ciphers that were proposed to constrained devices.

1. **Trivium (2006) [126]:** It uses 80 bit key, 80 bit initial Vector and 288 internal state. The aim was to design hardware-oriented binary additive stream ciphers which are both efficient and secure. The additive stream as the authors proposed takes as

input a k-bit secret key K, and an n-bit IV. Then, the cipher is requested to generate up to 2^d bits of key stream $\approx_t S_K(IV, t)$ where $0 \leqslant t \leqslant 2^d$, and then a bitwise exclusive OR of this generated key stream with the plaintext will produce the ciphertext.

2. **Enocoro-80 (2008) [189]:** It uses an 80 bits key, 64 bits of Initial Vector, and 95 bits of internal state. Enocoro-80 can be implemented with 2700 gates in ASIC. The implementation results were comparable to other ciphers selected as the eSTREAM Profile 2 candidates (hardware oriented ciphers). As a result, the design is suitable for software and hardware purposes.

3. **MICKEY v2 (2008) [164]:** It takes 80, 128 bits key and an Initial Vector of 0-80, 0-128 bits and the initial state is 200, 320 bits. MICKEY is an abbreviation of 'Mutual Irregular Clocking KEYstream generator', and this resembles the original design concept. The algorithm is based around two registers R and S, each of which has two modes of clocking selected by a control bit. The MICKEY family of algorithms was designed in response to the ECRYPT 'Call for Stream Cipher Primitives' in 2005 stream ciphers intended for use on resource-constrained hardware platform.

4. **A2U2 (2011) [240]:** It uses 61 bit key size, an Initial Vector of 64 bits and internal state of 95 bits. The lightweight cryptographic primitive has taken into consideration the extremely resource limited environment of printed ink tags, to develop a cipher that can be implemented with less than 300 gates, with the added benefit of high throughput provided by stream ciphers.

5. **Sprout (2015) [325]:** Sprout uses 80 bits as a key, and 80 bits of IV. The internal state is 288 bits. A new method for reducing the internal state size of stream cipher registers has been proposed in FSE 2015, allowing to reduce the area in hardware implementations. Along with it, an instantiated proposal of a cipher was also proposed: Sprout. The authors aim at reducing the size of the internal state used in stream ciphers while resisting to time-data-memory trade-off (TMDTO) attacks. They propose to this purpose a new design principle for stream ciphers such that the design paradigm of long states can be avoided. This is done by introducing a state update function that depends on a fixed secret key.

3.2.4/ Dedicated Authenticated Encryption Schemes

Some of the famous ciphers that were proposed upon the call of CEASER or NIST competition are listed below.

1. **ACORN (2016) [352]**: ACORN uses 128 bits key, 128 Initial Vector and 293 bits as internal state. The operations used are just bit-wise exclusive OR, bit-wise AND, bit-wise NOT and concatenation.

2. **ASCON (2016) [345]**: Ascon has been selected as the primary choice for lightweight authenticated encryption in the final portfolio of the CAESAR competition (2014–2019) and is currently competing in the NIST Lightweight Cryptography competition (2019).

3. **SAEAES (2018) [374]**: Is a family of authenticated encryption algorithm developed by Mitsubishi Electric Corporation and The University of Electro-Communications,

and submitted to Lightweight Cryptography Project by National Institute of Standards and Technology (NIST). It has a minimum state size since the state size equals to a block size of a block cipher. There is no need for an inverse to do decryption. Besides, only XOR is needed in addition to a block cipher encryption and it is an online cipher, i.e. a data block is processed only once.

3.2.5/ LIGHTWEIGHT HASH FUNCTIONS

It is harder to design and implement a lightweight hash function than a lightweight ciphers. For sure, they normally require a larger internal state which is applicable on desktop computers, but this would be costly on a limited device. For example, Sha-3 uses a 1600-bit internal state which shrinks the 64-bit block of most lightweight block ciphers. However, we list some of the lightweight hash functions below.

1. **Armadillo (2010) [214]**: It produces a digest with 80, 128, 160, 128 bits for 48, 64, 80, 128 bit block respectively and has an internal state of 256, 384, 576, 768. With fully serial architecture the authors obtained that 2923 gate equivalents (GE) could perform one compression function computation within 176 clock cycles. Note that a gate equivalents is a unit of measure which allows specifying the relative complexity of digital circuits.
2. **Spongent (2011) [236]**: It produces a digest of 80, 128, 160, 224, 256 bits and the block size is equal to 8, 16 bit. For the internal state it is equal to 88, 136, 176, 240, 272 depending on the digest desired. Its smallest implementations in ASIC require 738, 1060, 1329, 1728, and 1950 GE, respectively. The design is based on a PRESENT-permutation and this primitive provided the authors with confidence in its security with respect to the most important attacks.
3. **Blake2s/b (2013) [289]**: It produces 8-256, 8-512 digest with a block size of 512, 1024 bits and an internal state 512, 1024. BLAKE2b is optimized for 64-bit platforms, and BLAKE2s for smaller architectures. On 64-bit platforms, BLAKE2 is often faster than MD5, and it provides security similar to that of SHA-3: up to 256-bit collision resistance, immunity against length extension, indifferentiability from a random oracle, etc.
4. **Quark (2013) [288]**: Quark produces 136, 176, 256 bit digest and takes an input block size of 8, 16, 32 bit. The internal state is 136, 176, 256 bits. Quark can be used for message authentication, stream encryption, or authenticated encryption as well. The hardware evaluation shows that Quark is a great competitor to the previous lightweight hash functions. For example, the lightest instance u-Quark provides at least 64-bit security against all attacks (collisions, multi-collisions, distinguishers, etc.) and fits in 1379 gate equivalents.

3.2.6/ LIGHTWEIGHT CRYPTOGRAPHY USED AND GENERATED BY GOVERNMENTS

Some governments aimed at designing their own lightweight ciphers. They are often published by national standards. But some information regarding them are disclosed and are not shared to public. Below, we list some of these ciphers.

1. **DES (1999) [65]:** The Data Encryption Standard (DES) was released by U.S. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory. The exact generation of the Sbox remains a mystery although it is published. It uses 56 bits key, 64 internal state, and 16 rounds.
2. **ZUC (2011) [258]:** It is a stream cipher designed by the Data Assurance and Communication Security Research Center (DACAS) of the Chinese Academy of Science. It takes 128 bits key, with an internal state 560 bits and an Initial Vector IV of 128 bits. Also, although it was published, still there is no cryptanalysis from the original designers is published.
3. **SPECK (2013) [301]:** Speck is designed by the National Security Agency (NSA). It uses 64-256 bit key size, 32-128 bit block size and runs for 22-34 rounds. Although it was published, the security analysis for Speck was disclosed.
4. **SIMON (2013) [301]:** SIMON is designed by the National Security Agency (NSA). It uses 64-256 bit key, block size of 32-128 internal state and 32-72 round number. The disclosed information that was not published for Simon is the security analysis done upon it.

3.2.7/ LIGHTWEIGHT VS ULTRA-LIGHTWEIGHT SCHEMES

A large number of algorithms have been proposed to be in the lightweight category. They all operate in a different manner but they have one thing in common: being efficient in less powerful devices. The term lightweight can be defined in different attributes. For example, when taking the hardware constraints, Gate Equivalent (GE) is mainly used as a metric to measure how physically the circuit that implements the cipher is. The throughput is is measured in bytes per second which corresponds to the amount of plaintext being processed per time unit. Another metric is the memory used in the cipher, for example, storing the key and the full internal state and then performing one round of the cipher in i clock cycle (one round per a clock cycle). Also, related to hardware metrics, energy and power efficiency is one of the new criteria to be considered [326] Latency can be also added to the list which is a crucial metric when taking medical health as an example. As for the software measurement, the RAM needed, the code size, and the throughput (bytes/CPU-cycle) are all to be considered. Recently, one of the most famous evaluations that was done to evaluate some of the lightweight algorithms was FELICS [331] which stands for the ”Fair Evaluation of LIghtweight Cryptographic Systems”. Additionally, many researchers tend to set some metrics to hardly split between these two terms. In [368], the authors tend to propose that Equivalent gates is a good metric to specify whether an algorithm is light or ultra-light. They state that 1000, 2000, 3000 logic gates stand for ultra-light, low-cost, and lightweight ciphers. While in [356], the authors foresaw that splitting the algorithms into IoT ciphers and ultra-lightweight. They state some metrics that can specify the latter, for example, in ultra-lightweight algorithms, the security can have an excuse for not being at its top, since the main objective is not drained out of battery. The block size in their opinion should be 64 bits or more, and the key is at least 80 bits, besides using a volatile memory. While for IoT devices, the main objective is security, therefore, block size should be 96 bits (minimum), and the key is 128 bits at least. One algorithm or one suite should be available for IoT despite of its heterogeneity, since they are all connected. They also preferred the type of the cipher used to be either block or sponge.

According to our point of view, the most important criteria in any lightweight or ultra-lightweight is the execution time and the number of rounds. A trade-off must be made to reach an optimal number of rounds. Avoiding the use of float points, expensive operations (like matrices multiplication) and sticking to simple operations, are all factors that can render the cipher lightweight and suitable for these tiny targeted devices. In [375] we proposed using only a one round cipher that is suitable for IoT devices. the operations were simple and the cipher was proved to meet the cryptographic desired primitives. Additionally, in [376], two rounds were enough to reach the avalanche effect desired and the level of security wanted. The main change in our proposals is the induction of dynamicity principle in the cipher. The key and the cipher primitives will be calculated from a Master key which makes the cipher resilient to most of the attacks. A dynamic approach can increase the level of security, protect against physical attack and reduce the number of rounds.

3.3/ CONCLUSION

This chapter comes to summarize most of the State-of-The-Art related to lightweight cryptography. Starting by the definition of lightweight cryptography, we tended to shed the light on the importance of this field today. Using all the embedded sensors, devices, and the new evolution of the Internet of Things, mandated the need for security solutions that can cope with this tremendous change. IoT has been a life changing concept where everything is connected to everything and new set of challenges has arisen. Explaining the most famous standard AES was the beginning of this chapter to show that this cipher is expensive in terms of memory and operation for the targeted devices. Many researchers have worked in the domain of lightweight cryptography, and we state several works that caught the interest of both academy and industry. Lightweight cryptography has been used in block ciphers, stream ciphers, hash functions, and even has been used in governments like the United States. All of this shows the importance of this field in the platforms that exist today. We listed most of the block ciphers and showed that each proposed cipher has been either attacked, or is even heavy in terms of computation. We can come to a conclusion that having a new cipher, which we chose to be a block cipher, can be of great importance. After that, we state different metrics that can be used to differentiate between lightweight and ultra-lightweight ciphers and state our point of view in the last few lines. To wrap every thing up, we can say that lightweight cryptography is one of the most important fields today that attract both academy and industry.

III

CONTRIBUTION

4

INVESTIGATING VANET/IOT

4.1/ INTRODUCTION

The Internet of things has become an emerging paradigm. It is smartly changing the various existing research areas into new topics, starting with smart industry, smart health, smart houses and reaching smart transport. Intelligent Transportation Systems (ITS), the heart of the new revolution of smart transport, has evolved from the well-known Vehicular Ad-hoc Networks (VANETs) to become the Internet of Vehicles (IoV). In fact, the increase in the number of vehicles and the newly born technologies have stimulated the new Internet of Vehicles (IoV) or the Internet of Cars. In general, IoV aims at ensuring better traffic efficiency and reducing road accidents. However, due to different limitations and issues, both technologies IoV and VANET suffer from different security and privacy issues. In fact, they are both vulnerable to various types of security and privacy attacks that may result in life-endangering situations. As a result, several solutions were presented to achieve the required levels of security and confidentiality.

Having a universal network connecting all the available heterogeneous networks has become one of the great challenges researchers are interested in. This fact comes from the highly growing number of everything: smartphones, vehicles, appliances, laptops, tablets, sensors used in the daily life, etc. This global network is nothing more than what is commonly referred to as the ”Internet of Things”. In IoT, the inter-operability among the heterogeneous devices is the major objective. In fact, Internet of Vehicles (IoV) is one of the main topics in IoT. IoV has evolved from what has been known as Vehicular Ad-hoc Network (VANETs). VANET is a special type of mobile ad-hoc network used for communication between vehicles and roadside units. It's objective is to improve road safety, traffic management and congestion monitoring. The great change in vehicular networks was initiated in 2002, when researchers investigated the use of VANETs to reduce safety problems and to ensure more comfortable driving. In Europe for example, several automobile manufacturers have had the courage to carry out a real inter vehicle communication such as **Audi, BMW, Fiat, Renault**. These companies have cooperated to create a *Car2car Communication Consortium (C2C-CC)* organization [197], dedicated to inter-vehicle problems and issues. In addition to security applications, there are other applications for VANET such as infotainment, new payment strategies and insurance billing using advanced wireless access technology enabled in vehicles with or without the help of roadside units. However, despite all efforts to ensure that VANET gains investor and commercial interest [338], VANET has so far failed to achieve this objective. The strong

growth in the number of vehicles on the road is much greater than the capacity of the VANET technology. More than 125 million passenger cars with embedded connectivity are forecast to ship worldwide between 2018 and 2022. Currently, the connected car market is strongly aligned with 2G/3G networks, according to Neil Shah, research director at Counterpoint. However, he said that it is “moving swiftly” to 4G LTE connectivity, with the technology forecast to be installed in nearly 90 percent of connected cars by 2022 [373]. As seen, the purely ad-hoc nature of VANET [287], the lack of cloud computing and advanced computing despite all the ongoing attempts to integrate this functionality [328] and the lack of connectivity between the vehicle and personal devices [350] have meant that VANET has eclipsed its value. In addition, the main objective of VANET is to ensure driver safety, but safety solutions have not yet been developed at the time of writing this paper. Security has been one of the greatest challenges for VANET, and that's why researchers are trying to find the best reliable security solutions that can be used in IoV, since it has wider abilities [358] and more convincing arguments. IoV will enable the exchange of information between the vehicle and its surroundings using different communication means. Integrating the Internet of Things with VANET will create a new integrated network to support new applications, as intelligent traffic management, intelligent vehicle control, new information services [333], etc. IoV will enable the vehicles to be continuously connected to internet making it easier to provide information for these different services. Information is exchanged between the vehicles them selves, the passengers, the infrastructures parties, drivers, different sensors and electric actuators. In fact, this is the main difference between these two technologies, since IoV focuses more on the interaction between vehicles, humans and the available infrastructure. As a result, research in the field of IoV is currently becoming extensive and very active as it involves several axes at the same time, namely: wireless communications, protocols for physical and MAC layers, routing protocols and security, all with the aim of ensuring safe driving for drivers, and a better future for the IoV.

4.1.1/ MOTIVATION

The main contribution of this work can be divided into two major points. First, since VANET evolved into the IoV, it is important to indicate the differences between these two platforms and what are the main motivations behind this evolution. The second contribution is related to the security of such systems. Whether talking about VANET or IoV, using ad-hoc wireless communications, or 4G/5G communications, these platforms are sensitive to a large number of threats. Unreliable multi-hop transmissions, willful intermediate packet forwarding, and sharing specific personal data (location information or any sensitive messages) will certainly require a specific level of security. Therefore, the practical benefits of VANET/IoV could be mitigated in the absence of appropriate safety systems and could have a negative reverse effect on traffic and drivers' safety. Thus, the main requirement to ensure is the security of these heterogeneous systems where several security requirements are needed to resist different kinds of existing attacks [320, 192, 310, 317, 313, 308, 247, 285, 387, 295]. Shortly, this chapter summarizes and studies the various cryptographic/non-cryptographic schemes that have been separately proposed to secure VANET/IoV. Moreover, the existing security challenges for these schemes are well presented. Almost all recent existing primitive security solutions are analyzed for each threat jeopardizing the safety of these platforms. Each attack usually aims at affecting at least one of the following security services: availability,

authentication, integrity, confidentiality, privacy, and non-reputation. **Attacks are classified according to their security impact as well as the corresponding layer(s) in the Internet protocol stack layer they could affect**. In fact, a better classification of the existing attacks, threats on different network layers, and their countermeasures would allow researchers to find new and more effective security measures.

In this chapter, Section 4.2 is devoted to present an overview of VANET and IoV, stating the reasons that were behind this evolution and the main differences between them and the main challenges faced. Then, in Section 4.3 shows the applications and standards deployed in these platforms. Then, in Section 4.4, security issues are presented, where the attacks/attackers are both classified and a risk analysis is shown, then modern security layers and characteristics of communication types are described. After that, in Section 4.5, the different existing security architectures are shown. Finally, a conclusion is drawn in Section 4.6.

4.2/ Background, Motivation and Overview

In this section of the chapter, an overview of the conventional VANET is presented, followed by the motivations that encouraged the growth of the new platform IoV which will also be presented as a heterogeneous vehicular network.

4.2.1/ VANET's architecture

Vehicular Ad-hoc Network (VANET) is a promising area of research and development as it has remarkable role in improving safety of vehicles on road, efficient traffic management, and providing comfort to commuters in an affordable way. VANET has three main entities which are described as follows and are demonstrated in Figure 4.1:

1. **OBU:** An On Board Unit is equipped with each vehicle to provide wireless communication, allowing it to communicate with other neighboring vehicles to share traffic information and road conditions to ensure the global safety.

2. **RSU:** A Road Side Unit is immobile, not fully trusted, and subordinated by the Trusted Authority (TA). It is used to exchange information with TA and OBUs and can be compromised by physical attacks.

3. **TA:** A Trusted Authority, the registration of immobile RSUs and mobile OBUs is done by TA. It requires sufficient storage and computation capability to enable it to issue the main keys of the network.

These stations will communicate using an infrastructure specialized for VANET and, as far as we know, cars are being sold with the ability to communicate. In addition to the OBU stated previously, there are other components that are also added into the smart vehicle's system. For example, a GPS (Global Positioning System) is used for navigation, sensors (ladar and radar) used to detect objects at a certain distance, Event Data Recorder (EDR) which is a computing unit that can ensure the process and storage of data, a unique ID like the electronic license plate, a wireless transceiver that provides V2X communication

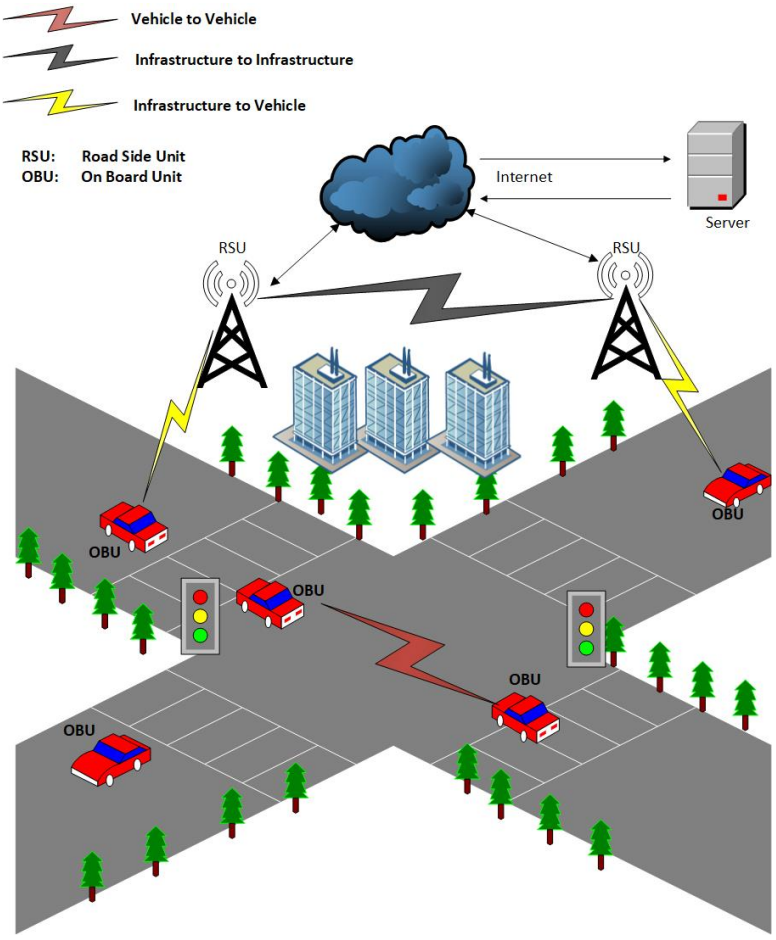


Figure 4.1: System architecture in VANET.

according to a standard; etc. In short, the smart vehicle is equipped with a communication system, a computing system, and a recorder box that records all the events exactly as a black box does in an airplane. In addition to all the aforementioned devices, each vehicle is equipped with a Tamper-Proof-Device (TPD) to store the secret information (private key) and is responsible for signing outgoing messages. To do that, a TPD contains a set of sensors that can detect hardware tampering. Once a tampering is detected, TPD removes all the stored keys, to prevent them from being compromised [154]. Moving to the communication side, vehicles communicate by node-to-node communication, where nodes establish connections with other nodes to exchange information in a short period of time. Communication in a VANET can occur through three kinds of vehicular communication methods [207] which are **vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and Infrastructure-to-Infrastructure (I2I)** as shown in Figure 4.1. V2V communications can be realized by employing IEEE 802.11p ad-hoc Network [175]. V2I communications are only based on ad-hoc communications (between the Vehicle and Roadside) or on generic wireless access network based Wi-Fi. In fact, V2I communication are less vulnerable to attacks and they require more bandwidth than V2V communication. This kind of connection requires a minimum lag and a low bit error rate. For this reason, it requires a reliable peer-to-peer channel, denoted by **Dedicated Short Range Communication DSRC** channel [143], that is presented by the **Federal Communication Commission**

which allocates a 75 MHz of licensed spectrum for DSRC in US (30 MHz were allocated by the European Commission), and is used now by the **IEEE 802.11p**, enabling a high data rate, and a short-range communication with a minimal latency. Finally, I2I communications interconnect RSUs between each other and RSUs to central(s) and this is done via the Internet domain.

4.2.2/ Motivations to launch Internet of Vehicles

Although VANET's aim is to enhance the safety of drivers with increased efficiency, the industry's interest in it was less than expected. In general, the commercial efforts in VANET were not enough [338]. Starting from the pure ad-hoc architecture that VANET has, the vehicle will lose all the services given by the network directly when it is disconnected. The collaboration with other networks is not an available feature in VANET [367]. In addition, commercial applications are not available in VANET due to the absence of continuous Internet connectivity [287]. In case of Internet loss, and due to the ad-hoc architecture, vehicles are not able to communicate with the driver's or the passenger's devices. In a world of Internet of Things, this feature will only affect the existence of VANET. Speaking of IoT, big data, and intelligent decisions, the common exchanged terms: edge computing, fog computing, or cloud computing are also not available in the current VANET architecture. And efforts are still being made to find a solution in spite of all the current challenges [359]. For these reasons, IoV was found to be more reliable and realistic in the big data era. Moreover, approximately 1.35 million people die each year as a result of road traffic crashes, and road traffic injuries are the leading cause of death for children and young adults aged between 5-29 years [378]. Due to these enormous numbers of deaths, there is an insistent need to start with an effective new solution based on safety applications without the need for continuous user intervention. A more reliable vehicular communication can be provided by IoV, thus decreasing the large number of road casualties. Finally, when talking about IoV, this opens the market to new demands. In fact, connected cars will turn into an increase in generated-revenue. Revenue projections from connected cars range from 40 billion dollars to worth of 100 billion dollars a year by 2020. Car manufacturers will benefit from connected vehicles and mobility services, but other industries are also in the process to benefiting from them. Mobile Network Operators will be the first to benefit from the connectivity required by IoV. In addition, the in-car technology will have the lion's share as new devices will be needed in the vehicles, new products and services will be adapted specifically to driving scenarios. Also, cloud services can help their businesses adapt to the accelerated development cycles and growing customer demands for connected cars. In addition, there will be a high demand for a higher processing power, thus the processor manufacturing will also have a new target. For example, currently, the NVIDIA Tegra X1 mobile processor for connected cars, used to demonstrate its Drive CX cockpit visualizations, can handle a trillion floating-point operations per second (flops) [342, 377].

4.2.3/ IoV overview

Internet of Vehicles has become a special application of the Internet of Things. It will make drivers enjoy a safe, convenient and comfortable driving experience. IoV is especially important for autonomous vehicles as they can spontaneously communicate with other

cars around them. This type of communication allows early notices of braking, changing lanes or turning and helps ensure smooth and safe transportation between autonomous vehicles. Cars are enabled cars with modern electronics and integration of the information to help maintain traffic flow, and to perform more effective fleet management and accident avoidance. The electronics used include special sensors, GPS, entertainment systems, brakes and throttles. There are five types of communication in heterogeneous IoV which can be summarized as following and as demonstrated in Figures 4.2 and 4.4:

- V2V: Vehicle to Vehicle- using IEEE WAVE
- V2R: Vehicle to Roadside unit- using IEEE WAVE
- V2I: Vehicle to Infrastructure of the mobile networks- using WIFI/4G LTE
- V2P: Vehicle to Personal devices (Laptops, smartphones..)- using CarPlay/NFC
- V2S: Vehicle to Sensors- using Ethernet/MOST/Wi-Fi

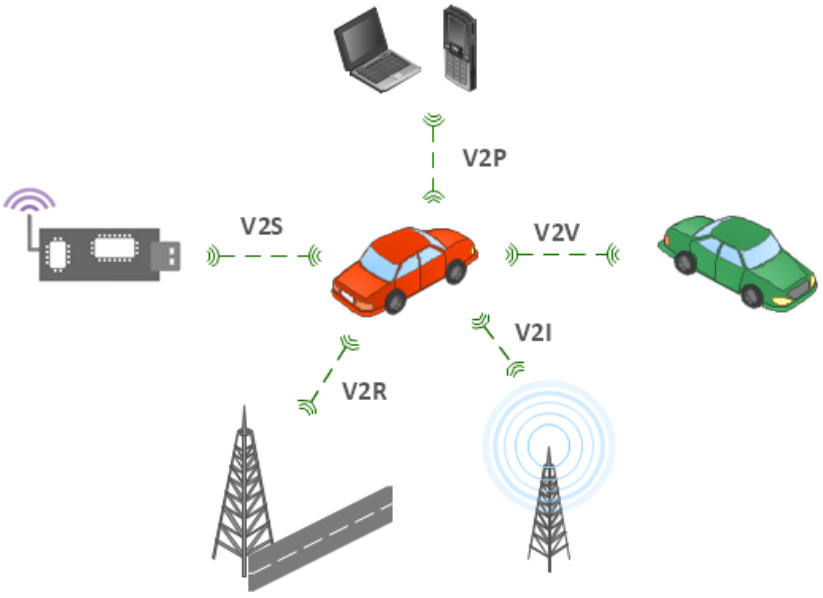


Figure 4.2: Types of vehicular communications in IoV.

A global network is enabled using Internet and other heterogeneous networks. The network includes IEEE WAVE for V2V and V2R, 4G/LTE and Wi-Fi for V2I, CarPlay/NFC for V2P, and MOST/WI-FI for V2S. It is obvious that due to the heterogeneous network environments in IoV, different wireless access technologies are utilized to establish connections. The vehicular networks are represented by different wireless access technologies (see Figure 4.3). The V2V and V2R networks represents vehicular communications through WAVE/DSRC. The V2I network demonstrates the vehicular communications through Wi-Fi or 4G/LTE [379]. The V2P network symbolizes the vehicular personal device communications using CarPlay of Apple or Android system of Open Auto-mobile Alliance(OAA) or Near Field Communication (NFC). The V2S network represents in-vehicle sensor communications through Ethernet,Wi-Fi or Media Oriented System Transport (MOST) [85]. This will add complexity to the architecture but will increase

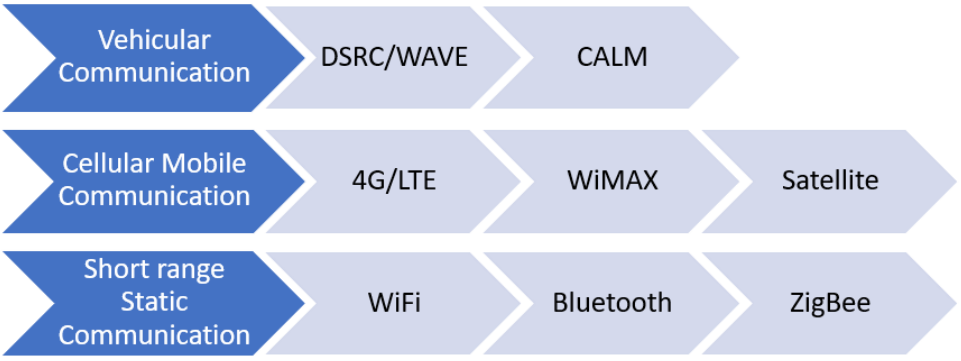


Figure 4.3: Wireless technologies for IoV applications.

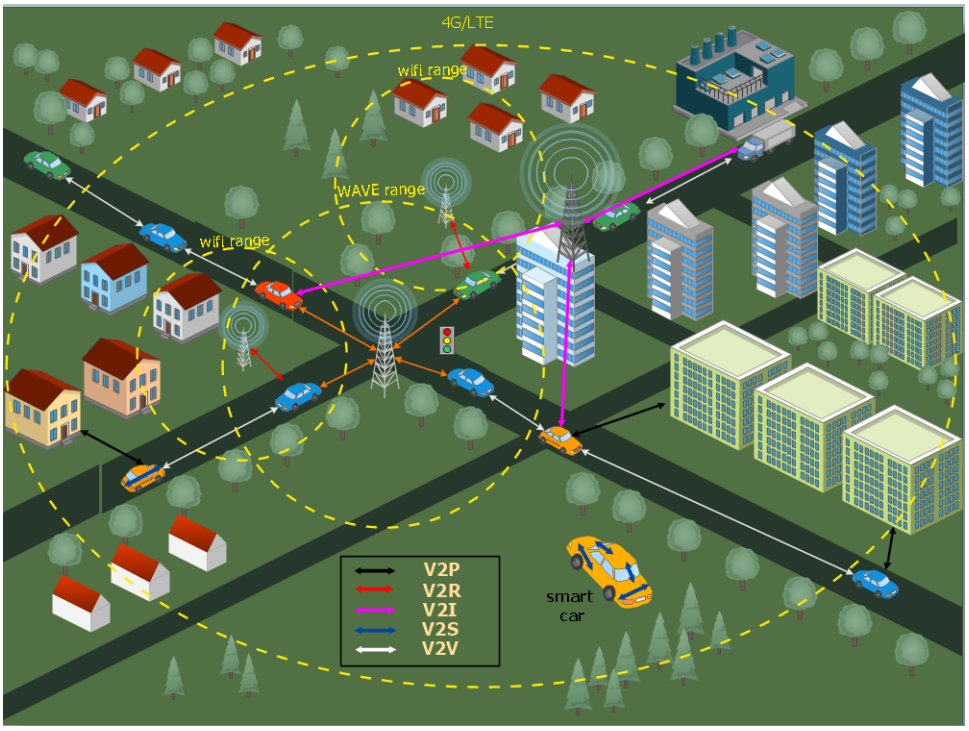


Figure 4.4: IoV and heterogeneous networks.

the interest in all industrial markets on contrary to VANETs. IoV will have a significant importance to supervise different vehicles with the intervention of all different available networks. It will provide a more reliable platform for Internet and multimedia applications related to safe driving.

4.2.4/ CHALLENGES IN IOV/VANETS

IoV and VANETS have several open research challenges and issues that should be addressed to provide a real, effective and safe deployment for ITS applications. Employing ITS applications suffers from several obstacles and constraints that are discussed in the following:

1. **Big data:**A major challenge is the processing and storage of big data created in IoV due to the large number of connected vehicles. For instance, autonomous cars are expected to process 1 GB of data per second. Mobile cloud computing and edge computing; available in IoV; will play an important role in handling the big processed data.
2. **High mobility:** Since we are dealing with mobile nodes, the prediction of these nodes is a difficult task to fulfill in terms of location and specifying the directions [200]. The positions of nodes in IoV/VANET can join or leave a network quickly and in a very short period of time, hence, different topologies are investigated with every new node position, making the network topology very dynamic and subjected to frequent changes causing a continuous link breakage between nodes.
3. **Hard-delay constraints:** Information related to safety such as the location of other cars must be sent as fast as possible to avoid any collision, therefore, the network must be very sensitive to delays to avoid catastrophic results. Safety related applications mainly needs a real-time response. However, these real-time constraints make the applications vulnerable to Denial of Service attacks (DoS), therefore, detection of real-time attacks is critical when insiders evades existing protection mechanisms.
4. **Scalable network:** VANET/IoV can be applied in urban or rural areas thus the network's size is not limited to a defined area [394]. In addition, the number of vehicles is estimated to exceed 250 million by 2020 [335], and until today, no global authority has provided the security for such large systems. A cooperation between worldwide local authorities is needed to achieve the standardized authority.
5. **Using wireless Communication:** As explained, the nodes communicate with others by wireless communications, so here comes the major role of security to ensure the safety of the information. In fact, data can be disseminated by the vehicles' communication, making the network vulnerable to attacks as bogus information attack.
6. **Different types of communication modes and technologies**: As stated earlier, there are different types of communication modes, so the connected vehicles must support a wide range of communication technologies such as IEEE 802.11p, Wi-Fi, Bluetooth, 4G/LTE, etc. Therefore, the vehicle must be equipped with the convenient hardware/software to support these heterogeneous platforms.
7. **Ensure high level of security and privacy:** All the information will be sent to different parties, thus the integrity, authentication, and availability must be considered. In this context, security protocols must be implemented with low communication overhead due to time constraints, and low computation complexity to exchange quick and safe information. A trade-off between latency and QoS must be ensured and having a lightweight encryption scheme is necessary to be able to respond to all requests at the same time at once avoiding any lags that can cause catastrophic accidents.
8. **Network Management:** Due to the large scale of the vehicular network that consists of millions of vehicles, and generates a huge amount of data that must be stored and distributed across the network, an effective network management must be used to deal efficiently with the network size and network produced data [185].
9. **Localization system:** Ensuring safety property requires a reliable and very accurate localization system. Normally, VANET/IoV uses GPS to enable the localization

process. But satellite-based positioning systems are not always available, especially when passing through tunnels, which makes the system vulnerable to several types of attack such as spoofing and blocking attacks. To deal with this problem, a number of localization techniques have been investigated such as Map Matching [149], Dead Reckoning [151], and Cellular localization [166]. Until now, there is no technique that can meet all of VANET/IoV requirements, such as time sensitivity, availability, and reliability. Here comes the need to build a reliable localization system, whilst satisfying all the critical points.

10. **Spectrum issues:** V2V communication system is intended to be used for at least 20 years and within this time the spectrum availability has to be guaranteed. In the US, the FCC has allocated 75 MHz of spectrum at 5.9 GHz (from 5.850 to 5.925 GHz) for C2C and C2I communications. VSC and VII Consortium agreed that the best technology available for the communications systems using this spectrum would be a derivative of IEEE 802.11, thus, the development of the IEEE 802.11p and ISO TC204. Unfortunately, a continuous spectrum of 75 MHz in DSRC band is not available in Europe. Hence, the Car2Car CC has proposed a subset of the US approach. The proposal allocates 2×10 MHz for primary use of safety critical applications at 5.9 GHz range (5.875 - 5.925 GHz) [174]. However, when the number of nodes sending periodic broadcasts is too large due to high traffic volume, some specific messages like emergency warning messages need a greater amount of time to be received, since bandwidth availability is minimal in wireless networks. Thus, the bandwidth must have a good management to prioritize the exchanged messages.

4.3/ APPLICATIONS AND STANDARDIZATION EFFORTS

4.3.1/ APPLICATIONS IN ITS (IOV/VANET)

The main target of ITS is to create a more efficient transportation infrastructure by employing vehicular communications that must improve **(1) road safety, (2) traffic efficiency and management,(3) comfort and infotainment, and (4) autonomous driving** in transportation systems as shown in Figure 4.5. In this context, several explored applications vary from a simple exchange of vehicle status to a complex large-scale traffic management.

4.3.1.1/ ROAD SAFETY APPLICATIONS

These kinds of applications are primarily employed to avoid dangerous collisions which may cause losses of drivers' lives. They provide drivers with all kind of messaging assistance to avoid collisions with other vehicles. Communicating and sharing information between vehicles and roadside units are two ways used to predict and avoid collisions. This shared information can be a vehicle's position, intersection position, car speed and distance heading. Moreover, locating dangerous locations on roads, such as slippery sections or avalanches can be easier. These applications can be classified into two classes:

- "Driver Assistance Applications (DAA)" inform and assist drivers to avoid road dangers or accidents. Three applications are being standardized by ETSI for DAA:

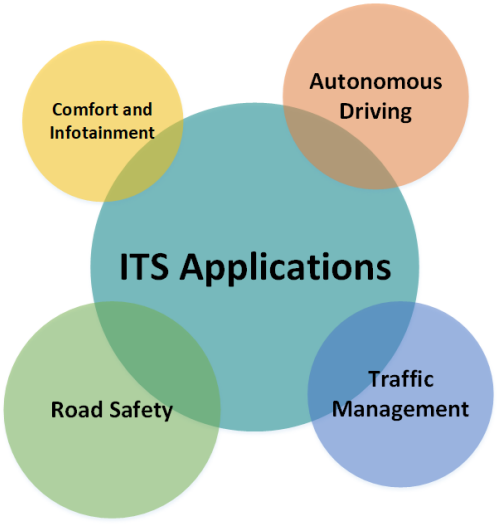


Figure 4.5: ITS main applications.

  - Cooperative Awareness Applications (CAA) [234];
  - Longitudinal Collision Risk Warning (LCRW) [389];
  - Intersection Collision Risk Warning (ICRW) [388].

- "Actions on Vehicle Applications (AVC)" that can provide necessary information for vehicles' systems to avoid or reduce accidents (Lane change assistance, pre-crash sensing/warning, emergency electronic brake lights, stationary vehicle warning, control Loss warning [393]).

4.3.1.2/ Traffic efficiency and management

Traffic efficiency applications improve and facilitate the management of the traffic flow and provide a cooperative navigation. Typical examples of these types of applications are (1) Speed limit navigation to help the driver control the speed of his/her vehicle for easy driving and to avoid unnecessary stopping, (2) Traffic information and recommended itineraries provisioning to enhance the traffic efficiency by managing the navigation of vehicles through cooperation among vehicles and road side units. These applications use several V2X messages like control messages such as Service Announcement Message (SAM).

4.3.1.3/ Comfort and infotainment

This kind of application aims at adding valued services. These services are mainly offered by service providers and are downloaded by drivers on their application units. There are mainly two kinds services available: (a) Cooperative local services which focus on infotainment that can be offered by locally based services such as point of interest notification, media downloading, local electronic business (b) Global Internet services which are mainly communities services (insurance and financial services, rapid management and parking zone arrangements, ITS station life cycle which is software and data updates).

ITS applications	Communication types	Exchanged messages	Mode of transmission	Security requirements
Road Safety	V2V	CAM	Broadcast	Authentication, Integrity, Privacy, Plausibility, Availability
		DENM	GeoBroadcast	
	I2V	CAM	Broadcast	Authentication, Integrity, Plausibility, Availability
		DENM	GeoBroadcast	
		SPAT	Broadcast	
Traffic efficiency	I2V	SAM + other messages	Broadcast	Authentication, Authorization, Integrity, Availability
Infotainment and comfort autonomous driving	I2V	SAM + EVCSN	Broadcast	Authentication, Authorization, Integrity, Availability
	V2I &I2V	-	Unicast & Broadcast	Authentication, Authorization, Integrity, Privacy, Confidentiality, Non repudiation

Table 4.1: Communication types, exchanged messages, mode of transmission and security requirements for different ITS applications.

4.3.1.4/ AUTONOMOUS DRIVING

Lately, autonomy has become a very topical issue. Cadillacs, BMW, Nissan can now drive themselves down highways hands-free, as long as the driver still pays attention and nothing out of the ordinary takes place; the new Mercedes-Benz S-Class can power through traffic circles, as long as you are actually doing the steering; and there are a few trials around the world of autonomous vehicles going on in severely controlled circumstances. In addition, augmented reality and virtual reality are both new concepts adapted in vehicles, where the user can enjoy his trip wearing a VR headset observing anything desired. In order to achieve that, new concepts should be adapted in vehicles to guarantee the safety of the driver [366]. This technology needs the vehicle to be equipped with ultrasonic sensors, 360 radars, 360 cameras, satellite systems, and the secure environment/standards that can enable this seem-less and autonomous driving.
ITS applications with their corresponding communication types, communication range, exchanged messages and required security services are presented in Table 4.1. Here is a list of the abbreviated exchanged messages:

- CAM: Cooperative Awareness Message. CAMs are sent by vehicles multiple times per second (typically up to 10 Hz), they are broadcast unencrypted over a single hop and thus receivable by any receiver within their range. Usually, vehicle's current position and speed are in CAM, along with other information such as steering wheel orientation, vehicle length and width, and brake state.

- DENM/DNM: Decentralized Environmental Notification Message. Transmission is triggered by a cooperative road hazard warning application, providing information to other ITS stations about a specific driving environment event or traffic event.

- SAM: Service announcement Message, this message is used at the application layer by RSUs to announce a service for vehicles such as an internet access for example.

- EVCSN: Electric Vehicle Charging Spot Notification.

- SPAT: Signal Phase And Timing, used to give the status of traffic controller and for other purposes.

4.3.2/ STANDARDIZATION EFFORTS

Two ITS standards are defined for many ITS communication architectures: IEEE Wireless Access in Vehicular Environments (WAVE) [125] and ETSI (European Telecommunications Standards Institute) organizations [241]. The architecture of each standard follows the seven layers of the OSI (Open System Interconnection) reference model as all the recent communication technologies such as LTE. A common part between both standards is composed of the physical and medium layers known as IEEE 802.11. Both ITS standards (IEEE 1609 [220] and ETSI TC ITS [241]) are very similar in several terms such as offered networking, application management functionalities, and security. The entire protocol stack of ITS standards consists of DSRC (Dedicated Short Range Communications) [244], the common part IEEE 802.11p [275], and WAVE (Wireless Access in Vehicular Environments) [210] or ETSI standards, which are described in the following.

4.3.2.1/ DSRC

The ITS network uses a specific frequency band between 5850 to 5925 GHz (75 MHz bandwidth), which is known as Dedicated Short Range Communications (DSRC) [244]. This band can be divided into seven channels of 10 MHz, numbered 178, 172, 174, 176, 180, 182, and 184 respectively. The CCH channel (Control Channel) corresponds to channel 178. The other channels are used for SCH channels (Service Channels). Two service channels (172, 184) are reserved for high Availability and low Latency, and for high power and public safety. In Europe the situation is different. The DSRC band in Europe is regulated by the ETSI, and 5 channels are used; CCH uses channel 180 and the rest (172, 174, 176, and 178) is used for SCH.

4.3.2.2/ WAVE

IEEE published in [125] the latest ITS standards fact sheets, which declare the WAVE IEEE 1609 family (Standard for Wireless Access in Vehicular Environments). They introduced different services and interfaces in addition to security architecture that should protect the WAVE stations from various attacks. In addition, operation in an ITS environment and establishment of an efficient and secure V2X communication are both guaranteed. The WAVE architecture of the different 1609 standards and their integration with the OSI reference model are illustrated in Figure 4.6. Let us indicate that WAVE standards define the basis for the implementation of a wide set of applications in the ITS that include the safety of vehicles, traffic management, automatic tolls, improved navigation, and several other applications.

4.3.3/ ETSI ITS STANDARD

A standard of ETSI is presented in [217] and describes the European ITS communication architecture and specifies the comparison with the traditional OSI layered model. This standard consists of four layers: Applications, facilities, networking/transport, and access in addition to two cross layers, which are security and management are illustrated in Figure 4.7.

Standard	WAVE definition
IEEE P1609.0	Architecture of WAVE
IEEE P1609.1	Resource Manager.
IEEE Std 1609.2	Security Services
IEEE Std 1609.3	Networking Services.
IEEE Std 1609.4	Multi-Channel Operations.
Draft IEEE P1609.5	Layer Management.
Draft IEEE P1609.6	Remote Management Services.
IEEE Std 1609.11	Data Exchange Protocol Over-the-Air
IEEE Std 1609.12	Provider Service Identifier Allocations (PSID).

Table 4.2: List of of standardized protocols of WAVE

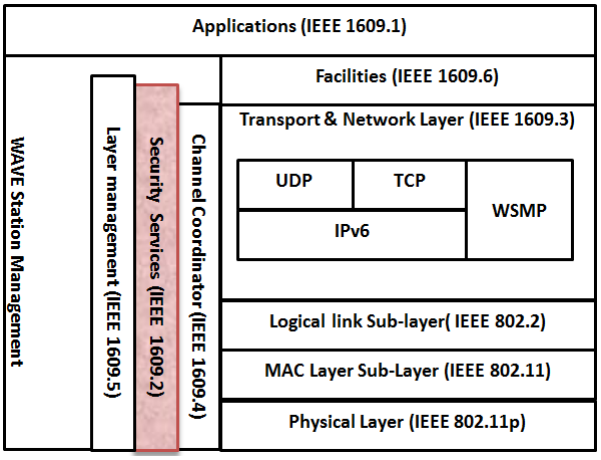


Figure 4.6: WAVE standards for ITS Layered Architecture for V2X Communications (US)

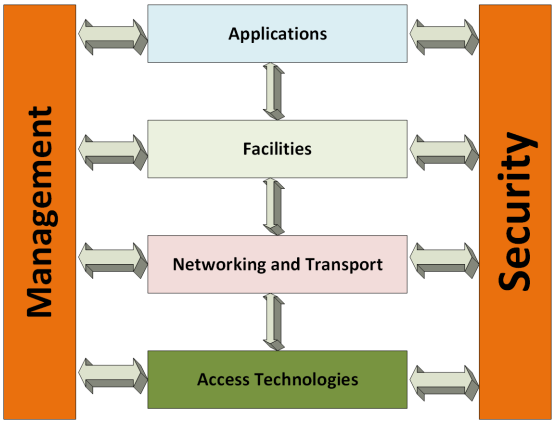


Figure 4.7: ETSI standard architecture.

- **Application layer**: It is responsible for the execution and the implementation of one or several ITS applications such as road safety and efficiency.
- **Facilities layer**: Its objective is to be a middle communication layer between application and network layers.

- **Networking & transport layer**: It ensures the data transport between source and destination stations of ITS. In fact, it can be composed of two parts: ITS transport and TCP/UDP connection and includes the support of GeoNetworking, IPv6 networking, TCP/UDP transport protocol, etc.
- **Access layer**: It makes it possible to ensure wired and wireless communication technologies that are available in an ITS station. The access technology used for Safety applications for ETSI is ITS G5 which appears as the European profile of IEEE 802.11p.
- **Management cross layer**: It manages the communications depending on the requirements of ITS applications and it manages the features of the whole ITS architecture layers.
- **Security cross layer**: Its objective is to provide the security services.

4.4/ ATTACKS/ ATTACKERS MODELING AND ITS RISK ANALYSIS

VANET/IoV were investigated for providing safe and fast rides, but because of the wireless network, several kinds of hackers can attack the system, degrade it and eventually cause accidents. Since the safety of people is involved, providing better security is mandatory. Vital information in ITS should be protected to prevent an attacker from modifying or deleting them. Secure transportation systems must also be able to determine the responsibility of drivers while maintaining their privacy [285]. Data exchanged through a vehicular network, information about the vehicles and their drivers must be secured and protected to ensure the reliable functioning of intelligent transportation systems [222]. However, VANET/IoV are known as a highly dynamic environment with short connection period duration that prevents the deployment of a complete and practical security solution. Ensuring the security in ITS can be considered as a complicated task and any security breach leads to critical and dangerous consequences. In fact, security breaches are likely to occur when using wireless media, dynamic network topology, high mobility, and diverse involved entities. In the following, the different security requirements and threats are described.

4.4.1/ PARTIES INVOLVED IN SECURITY

The different parties involved in the security of ITS system are:

- **The driver**: Drivers are receiving the information, so they are the most important element and their safety is a priority. Any wrong information sent to drivers can lead to their death.
- **The Road side Unit**: We can distinguish between normal RSU terminals, which operate in a normal way, and malicious RSU terminals.
- **The vehicle (OBU)**: The driver and the vehicle are both referred to at the same time. Two types of vehicles can be distinguished : normal vehicles that are found between the network nodes and operate normally, and ambiguous vehicles.

- **Third parties**: Third parties can be trusted or semi-trusted, and are responsible for RSU and OBU certificates and have the diverse secrets/public key pairs. They can be the regulators of transport, vehicle manufacturers, traffic police, and judges.
- **The attacker**: An attacker's target is to violate successfully the security of normal vehicles.

4.4.2/ ITS SECURITY REQUIREMENTS

To ensure a practical deployment of ITS, diverse security requirements must be reached to ensure safe driving and secure communication. Below, the requirements of VANET and IoV security are detailed.

- **Data confidentiality**: The sender station must be sure that the exchanged message can only be decrypted by authorized users. However, the confidentiality in ITS is not essential compared to other kinds of MANET network such as WSN, because safety messages should be shared [154]. However, several applications in ITS transmit sensitive important information that require confidentiality such as in [182] by using anonymous key pairs that can ensure privacy. In fact, the security in ITS requires a lightweight yet secure cryptographic solution.
- **Data integrity**: It ensures that the exchange of information is not changed during forwarding from the sender to receiver.
- **Authentication**: It can be classified into three sub-requirements: (a) user authentication to prevent Sybil attacks and prevent attackers from threatening the security of the system; (b) source authentication to validate that the messages are generated by trusted entities; and (c) location authentication to validate the relevance of the received information.
- **Privacy**: It is the most important security requirement, especially for ITS application since personal data are exchanged over wireless communications. An important requirement is to preserve the privacy of the driver against un-authorized observers. For that purpose, privacy should be ensured by protecting personal data and the design of ITS security solution must ensure this requirement with a lower latency. For example, if a node presents its certificate to one RSU at location x, then it presents the same certificate to another RSU at location y. Therefore, any attack observer can simply know that the owner of this certificate traveled from location x to y. For this reason, all private information of a node must be hidden and is only accessible by the Trust Authority (TA).
- **Availability**: Exchanged information should be processed and made available. These kinds of attacks are very dangerous for real-time applications since a small delay can make the message useless.
- **Traceability and revocation**: The malicious entities that are attacking the system must be monitored, in order to block them at the right moment. The trust authority should be able to trace the attacker and reveal its true identity. In addition, in case of a dispute or when a malicious entity is detected, the TA must abolish it and add its true identity to the blocking list.

- **Authorization**: It is necessary to define the rights and authorization of different entities (vehicle or infrastructure) for several applications to prevent entity attack.
- **Non-repudiation**: It is necessary to get the proof of the message originator for several applications such as in road safety where crucial information should be exchanged and could lead to dangerous consequences. It may be crucial in some cases (e.g. wrong information that causes an accident) not only to identify a sender but also to get the proof of the originator of the message (for accountability).
- **Immunity against physical attacks**: ITS entities should be immune against external attacks, such as availability attacks.
- **Scalability**: The term scalability implies that despite the fact that the activity volume gets expanded, there should not be any performance degradation or even network blackout, without changing the system components or protocols.
- **Delay constraints**: In some situations, the delivery of emergency messages on time is essential to preserve the safety of the driver.
- **Mobility**: As one of the characteristics of VANET is dynamic topology, a perfect mobility model is required to develop VANET environment effectively and efficiently.

4.4.3/ ATTACKER PROFILES

The network attackers' profiles should be specified when security issues are addressed in addition to the possible kinds of attacks. There are three categories of attacks: (1) active and passive, (2) malicious and rational (3) internal and external according to [154]. In the following, these categories are briefly elaborated.

4.4.3.1/ ACTIVE VS. PASSIVE

An active node is a node that can send messages to cause harm to different nodes or to a part of the network. Mainly, this attacker is authorized to operate in the network. On the contrary, passive nodes simply eavesdrop communications that occur between nodes in a network. Passive attackers do not have any authorization. They will monitor the network and try to find some information. Even though this will not cause any real damage to the network, the collected data can be used by the attacker for other attacks later on.

4.4.3.2/ EXTERNAL VS. INTERNAL

External attacker nodes are not authenticated in the network. In general, external attacks are less likely to occur in the network compared to the internal ones. An external attacker has the possibility to perpetrate confidentiality and availability attacks. Confidentiality attackers are those who secretly eavesdrop information without the awareness of legitimate nodes and try to collect any useful data about road users that can be useful for a future attack [162]. Another example of external attack is the availability attacks such as DoS. Several kinds of DoS can be perpetrated and their main goal is to jam the network with fake messages since users in the network will receive these messages and the network

will be unavailable. In contrast to the external attack, the internal one can perpetrate all sorts of attacks in the network and can be divided into two types, which are respectively, authenticated nodes and industrial ones. respectively.

4.4.3.3/ MALICIOUS VS. RATIONAL

Malicious attackers do not have any specific goal and are not looking for any specific result. Their attacks are only conducted because it can be done, with no purpose in mind. Their main purpose is to damage the network by different ways like transmitting false information to different vehicles in a specific geographical area [153, 256]. In contrast, rational attackers have a specific target and can be dangerous [248], they are unpredictable and follow the passive class such as confidentiality attacks.

4.4.4/ CHARACTERISTICS OF ATTACKS

Figure 4.8: Characteristics and Profiles of attackers

In order to ensure a safe ITS implementation, the attack characteristics should be studied and analyzed in order to provide a robust scheme [154, 170]. One can consider a malicious data attack as a situation in which a malicious node tries to convince other nodes to accept corrupted data. The attack is successful when a node accepts corrupted data from a malicious node. Generally, the attacks can be characterized by five elements [102] described as follows and shown in Figure 4.8:

- **Nature**: The type of attack that can be perpetrated corresponding to the nature of attacks, determines the technique that can be used by malicious nodes to harm the network or its nodes such as authentication or availability attacks (DoS, jamming, Sybil).
- **Target**: The distance between the malicious nodes and victim nodes determine the target of the attack. For example, in ITS, convincing the victim to accept or process false data can be successfully achieved if the malicious nodes are geographically close to them. A node in ITS-Application uses information from several closer nodes to make a decision [392]. For that, malicious nodes require several allies to convince

the other legitimate nodes about specific target information. Whereas, for the long distance, cooperation between the different malicious nodes is necessary to prove the sincerity of their information.

- **Scope**: The attack scope defines its corresponding area, which can be quantified from small to big. Additionally, the attacker always tries to reduce its limitation and extends its infected area [177], which will increase the number of victims.
- **Impact**: The impact of the attacker measures the amount of damage produced by an attack.
- **Capacity**: The goal is to determine the capacity of protection to prevent or at least reduce (if prevention is not possible) the corresponding produced damages from this attack.

The malicious node attacks can be classified into three categories:

- **Detected and corrected:** When victims realize the uncertainty of false data they received from corrupt target nodes and are able to correct them.
- **Detected but not corrected:** If victims are able to detect the attack while unable to correct the damage caused.
- **Undetected and uncorrected:** When all of the victim nodes cannot detect the attack [96, 105, 106], it can be considered as the worst situation. This can occur in certain contexts, for example, if a victim node has no contact with trusted nodes that can help to overcome the situation by verifying the false data.

4.4.5/ Classification of ITS Attacks and their corresponding solutions

ITS is vulnerable to different kinds of threats and attacks as any communication system. In contrast to wireless sensor networks, the energy problem is absent and additionally, an OBU has the ability to harmonize dozens of microprocessors, which gives an important capacity of processing and computing to the vehicle [285]. Encrypting the sensitive message (or the sensitive part of the message) can provide better robustness and resistance against the passive attack and ensures the user privacy. Therefore, the transmitted data is protected from any unauthorized access. On the other hand, it is also necessary to ensure that this data is only exchanged between legitimate parties and is not being altered at the intermediate nodes. Classifying attacks is the first step to have suitable security solutions. Mainly, security solutions can be split into two main branches: **cryptographic solutions and non-cryptographic solutions**. In [211] a classification of existing attacks have been proposed. Attacks are classified according to their target, whether they attack the vehicle or they attack the RSUs. As in [317], a cryptographic related classification is used and expanded to clarify the cryptographic solutions to VANETs security issues. In this paper, attacks are classified according to their impact on the existing modern cryptographic security requirements which are: **Availability, Authentication and Identification, Integrity and Data trust, Confidentiality, Privacy and Non-repudiation.** Each attack will be classified according to what criteria of those it affects, stating also the layer it performs on. To satisfy these security services, several methods are used. For example, cryptographic algorithms mainly use encryption/decryption algorithms that generally

include key generations and protocols to protect the exchange of shared data, hash functions that are widely used, digital signatures and many other tools. Attacks and their corresponding solutions are presented in the following.

4.4.5.1/ AVAILABILITY ATTACKS AND PROPOSED SOLUTIONS

Availability is the most crucial factor in transportation's security system. It means that the network is functional at any time to get useful data.

- **DoS attack:** One of the most dangerous attacks in availability is the Denial-of-Service attack (DoS). Another kind of DoS is Distributed Denial-of-Service attack (DDoS). The main intention behind DoS or DDoS attacks is to make a service unavailable and cause havoc rather than trying to breach the security perimeter of the target. In most cases, the methods of DDoS attacks aim at flooding the network and the results are always dreadful.
  -Layers targeted: Multi-layer attack (Physical, Data Link, Network, Transport, Application layer)
  -Solution: An efficient solution is presented to reduce the impact of DoS attacks, which consists in the use of bit commitment and digital signature based authentication mechanism [269]. Also, in [306] a distributed and robust approach is presented to protect the system against DoS attacks. Another way to fight against DoS attacks is to use ingress routers that verify the identity of packets entering into the domain, or to use the Route-based Filtering approach which relies on route information to filter out spoofed IP packets [93]. From a non-cryptographic point of view, to prevent layer 7 DoS attacks, employing an application firewall or proxy-based application delivery solution ensures the fast and secure delivery of an application. By preventing both layer 4 and layer 7 DoS attacks, such solutions allow servers to continue serving up applications without a degradation in performance caused by dealing with layer 4 or layer 7 attacks [111]. In the following, some other examples of intended DoS attacks are briefly described with their corresponding cryptographic solutions.

- **Jamming attack:** This occurs by sending a noisy high-frequency signal in a channel which will result in a lower SNR preventing the vehicles from communication [252] [225].
  -Layers targeted: It is based on producing an interference at the Physical Layer.
  -Solution: The effect of jamming for mobile ad-hoc networks can be reduced by using different techniques such as in [225] and [196]. To reduce the effect of this attack, the frequency hopping technique FHSS (Frequency Hopping Spread Spectrum) of the used standard OFDM [300] should be randomized by the hopping algorithm. Furthermore, a pseudo-random generator should be used for this purpose on top of modifying the existing standard. In some cases, it is simply impossible to defend the system against jamming as an experienced attacker may have the ability to flood all available network frequencies. If the major concern was about malicious jamming, an intrusion prevention and detection system may be the best option. At the bare minimum, this type of system should be able to detect the presence of an RPA (Rogue Access Point) or any authorized client device in the wireless network.

Table 4.3: Different types of availability attacks with their corresponding solutions.

Name of Attack	Communication Types	Proposed Solutions	Possible Reason(s)
Denial of Service	V2I/V2V	[269] [306] [93] [111]	OBU vulnerabilities,Insecure wireless communication channel
Jamming	V2I/V2V	[225] [196] [300]	OBU vulnerabilities,Insecure wireless communication channel
Sybil Attack	V2V	[265] [139] [104] [34] [123] [209] [298]	Flaws in routing table and unencrypted messages
Malware	V2V/V2I	[302]	Software flaw and weak message propagation algorithm
Spamming	V2V	[123] [334]	Software flaw and weak message propagation algorithm
Black-Hole	V2V	[357] [213]	Unencrypted backend communication channel
Gray-Hole	V2V	[73]	Unencrypted backend communication channel
Worm Hole / Tunneling	V2V	[205] [180]	Unencrypted backend communication channel
Sink Hole	V2V	[132]	Unencrypted backend communication channel
Greedy Behavior	V2V	[316]	Broadcast nature of messages via communication channel
Hardware Tampering	V2V	[181]	Physical access to vehicles

- **Sybil attack:** In [83], an attacker creates a large number of pseudonyms, and fools vehicles to think that there is a traffic jam ahead of them and forces them to tell

other vehicles that there is jam ahead, then makes them take an alternate route.
-Layers targeted: Data link, Network, Transport, Application and sometimes Physical Layer.
-Solution: To avoid this kind of attack, a Central Validation Authority (CVA) should be used to certify the parties in real time. This process of certifying the nodes can be direct or indirect. In the direct validation, any incoming node should validate itself by using the CVA to establish a direct connection. On the other hand, the indirect one enables an already accepted entity to credit an incoming entity. The certificates used here by the CVA are temporary [265]. In addition, deploying the distance bounding protocol as bit commitment and zero-knowledge [104], [34], [123] and [209] will definitely strengthen the authentication process. Moreover, another solution to reduce the effect of a Sybil attack is presented in [139] which consists in validating ambiguous nodes by using secure location verification. Alternatively, proof of work can be used to make Sybil attacks more expensive [298]. In [221], a Sybil attack is detected by using physical layer parameters such as the received signal strength and the angle of signal. In [203], Park et al. propose a time stamp based approach to detect Sybil attack in VANETs. Different defense mechanisms against Sybil attacks are proposed in [337] and they propose a new secure event-reporting scheme that is resilient to Sybil attacks. Let us indicate that Sybil attacks can affect the network authenticity as well.

- **Malware attacks:** Malware attacks [285, 387, 265], such as viruses, Spyware, Adware, Trojan horses, Logic bombs and Cookies, have the potential to cause serious disruption to its normal operation. Malware attacks are more likely to be carried out by a malicious insider rather than an outsider. These attacks may be introduced into the network when the vehicles or the roadside stations receive software updates.
-Layers targeted: Application Layer
-Solution: Software companies develop detection systems products at laboratories and keep track of new programs, analyzing them, putting the valid software in the whitelist and the malicious software in the blacklist. For the undecidable software, which is called the gray list, the scanners operate them in a controlled environment for more classification. When an analysis of a program in the gray list results in new malware, the company releases online updates for the new malicious software. Then, users can update their product databases by using remote access through an Internet connection. Signature-based and anomaly-based with artificial intelligence (AI) techniques were used to enhance their efficiency. Neural networks (NNs) have been adopted for their adaptability to environmental changes and their efficiency in prediction techniques [302].

- **Spamming attacks:** The presence of spam messages like ads heightens the risk of increased transmission latency, and therefore might cause accidents. The lack of centralized administration causes serious problems. It is difficult to deal with spams because of the lack of infrastructure [285, 387, 254].
-Layers targeted: Application Layer
-Solution: Naive Bayes, Clustering and Decision trees are being used to improve the detection and prevention of spams [334]. Also, using digital signatures of software and sensor is crucial so that only authorized nodes can send and receive data [123].

- **Black-Hole attacks:** In black hole attacks, a malicious node uses its routing protocol in order to advertise itself for having the shortest path to the destination node

or to the packet it wants to intercept [266]. Also, in black hole attacks, nodes can refuse to participate in the network or simply drop out. All network traffics are redirected to a specific node, a non-existent node, which causes data to be lost.
-Layers targeted: Network and Transport Layer.
-Solution: Black hole attacks could be detected by using a quality control chart [357]. Another solution was proposed consisting of checking the good forwarding of the traffic by an intermediate node based on the well-known principle which is the Merkle tree [213].

- **Gray-Hole attack:** It is somehow different from the black-hole attack and drops the data packets corresponding to specific applications [279] that are vulnerable to lose during routing.
-Layers targeted: Network and Transport layer.
-Solution: Using Intrusion Detection System as proposed in Ahmed, M. et. al. [73], where every mobile node carries intrusion detection system which monitors the whole network structure with in-built mechanism.

- **Wormhole and Tunneling attack:** A Wormhole attack requires two nodes at least to participate. It happens when an attacker A sends a false message to an attacker B who is technically far from him/her. This message shows to the neighboring nodes of B, that A is near them as well [107]. In this way, the interchanged control packets among them [205] cause to create non-existing roads according to their neighbors. Tunneling attacks are like wormhole attacks [285] but with one difference which is using the same network to initiate a private connection (tunnel) in contrast with Wormhole attackers that use a different radio channel for the exchanging packets. An additional communication channel (tunnel) is used by the tunneling attack which establishes a connection between two far nodes in the vehicular network.
-Layers targeted: Network and Transport layer.
-Solution: One solution is proposed in [205] by Safi et. Al which introduces a packet leashes method to defend against the wormhole attack. A leash [180] is any information that is added to a packet designed to restrict the packet's maximum allowed transmission distance.

- **Sinkhole attack:** The packets of neighboring nodes go through a malicious node, which can eliminate or modify the received packets before eventually re-transmitting them. Moreover, the Sinkhole attack can be used to mount other attacks as the Gray-hole and the Black-hole attacks [92].
-Layers targeted: Network layer.
-Solution: In [132], a new light-weight algorithm to detect sinkhole attacks and identify the intruder in an attack is proposed. They examined multiple suspicious nodes and concluded the intruder based on majority votes.

- **Greedy Behavior Attack:** It is when greedy or selfish drivers aim to use network resources for their own benefit. It can cause an illusion of traffic congestion in its neighborhood. The attacker may also persuade the neighboring vehicles that there is a congestion in a specific route, thus they will use alternate routes and this will grant him/her a clear path to his/her destination [142].
-Layers targeted: Manipulate specific Data Link layer parameters.
-Solution: In [316], a new detection algorithm for greedy behavior attacks is proposed based on a statistical method, linear regression and watchdog software.

- **Hardware Tampering:** Hardware tampering can occur at the manufacturing level or by other mechanical ways that manipulate the node physically [133]. If materials are physically damaged, communication is disturbed and becomes unavailable [253].
  -Layers targeted: Physical Layer.
  -Solution: One of the proposed solutions is using Trusted Platform Module (TPM) [181]. Here, a driver must perform a physical verification. Hardware tampering also includes sensor tampering which means alteration of the position, speed and the orientation of other cars by an attacker. In case of an accident, the responsibility will fall on the attacking node rather than on the attacker.
  All availability attacks and their solutions are summarized in Table 4.3.

4.4.5.2/ ATTACKS ON AUTHENTICITY AND IDENTIFICATION

Authenticity is considered as a hard challenge in ITS security where the legitimate nodes should be protected against different kinds of attacks. It enables the receiver to validate the origin of data received. In fact, available services should only be accessed by the authenticated nodes and any fragility in the process of authentication or identification leads to perilous consequences in the network. An outside or inside attack can be prevented by ensuring the authentication using a falsified identity [285]. Whenever a vehicle needs to join the network or needs any service to allow the access, first it should pass through the process of identification-authentication. Let us note that the term of "authentication" from a cryptographic viewpoint means both authentication and integrity. Even though in this part, the focus is on authentication attacks, in the following part, integrity attacks are described. A list of these attacks is described in the following:

- **Node impersonation attack:** In the impersonation attack, the attacker obtains the credentials for another legitimate vehicle in the network. Every vehicle has a network ID which allows to distinguish it among the other nodes [265]. The attacker can advertise fake routes to confuse others, forward a route message with false sequence numbers to delay other messages, and also is able to flood the network by DoS attacks. In many networks, a malicious node could obstruct proper routing by injecting false routing packets into the network or by modifying routing information.
  -Layers targeted: Network Layer
  -Solution: Use Secure Ad-hoc On-Demand Distance Vector (SAODV) which depends on (1) **Hash chains** to secure mutable fields of the messages (hop count information is the only mutable field), and (2) **Digital signatures** to authenticate the non-mutable fields of the messages [89, 159]. Another solution was proposed in [95], which is a Double Authentication (DA) scheme which provides authentication to the routing information data carried by the link state routing packets. Every router needs to sign the routing data twice with two different keys using a group keying scheme, which is based on one-way hash function. In addition, in [237], they proposed the first group communication protocol to allow vehicles to authenticate and securely communicate with others in a group of known vehicles.

- **Key and/or Certificate Replication attack:** Duplicate keys or certificates are used as proof of identification to create ambiguity. Therefore, this prevents the authorities from recognizing a vehicle.
  -Layers targeted: Network Layer

-Solution: Using certified and available keys will protect the exchanged data. In addition, checking the validity of digital certificates in real time via CRL (Certificate Revocation List) [154] can be used. Another method is to apply the cross certification that occurs between the different certified authorities in the security network [134].

- **Illusion attack**: It is also an attack against integrity and data trust. Sensors are placed in the network to generate false data [293] and the vehicle it self needs to deceive its own sensors. As a result, false data can spread the network. In this attack, the protection process of authentication is not efficient, since the attacker is already authentic. These fake messages can be exchanged normally in the network and are capable of changing the decision of the drivers.
  -Layers targeted: Application Layer
  -Solution: In [150], they developed a new model, called plausibility validation network (PVN), to protect against fraud messages in traffic safety applications. Also, the signature can be used to detect only authentic location data [269]. Another solution is to use a reputation score for safety applications to detect the malicious nodes [294].

- **GPS spoofing/position faking attack**: "Jamming just causes the receiver to die, spoofing causes the receiver to lie" say consultant David Last, former president of the UK's Royal Institute of Navigation. Here the attacker can change the geographical information retrieved by GPS satellites by producing stronger signals. Thus, drivers might think they are in the right place when they are not. Particularly in transportation systems, the location of information is a very critical point, it must be precise and true [387]. This attack is done when false location information is transmitted to the neighboring nodes. Locations and geographical positions of all vehicles in the network are maintained using the genuine GPS satellite. However, an attacker can use a GPS satellite simulator that is more efficient than traditional GPS satellite, and allows to produce stronger signals [99], to track the node locations. So, other vehicles believe that they are in different locations, which can potentially cause collisions. This threat poses a critical problem in the vehicular network. A successful GPS spoofing attack can open the door for other attacks such as the ones against applications which use the position of the node for identification.
  -Layers targeted: Application Layer
  -Solution: To avoid this attack, bit commitment and signature scheme can be used. These methods work with positioning systems that only accept authentic and real data location [269, 104, 209].

- **Timing attack:** The timing attack is to delay the transmission of messages with high requirements on propagation delay, and transmits them, e.g. after adding time preventing their treatment in a normal way. Some classifications such as in [251] and [265], also consider this category as a separate family of attacks.
  -Layers targeted: Transport Layer
  -Solution: This attack can be made inefficient by using the "time stamping mechanism" for packets of delay-sensitive applications. However, this proposition encountered the problem of time synchronization between entities [291].
  All authentication attacks and their solutions are summarized in Table 4.4.

Table 4.4: Different types of authentication attacks with their corresponding solutions.

Name of Attack	Communication Types	Proposed Solutions	Possible Reason(s)
Node impersonation attack	V2V	[89] [159] [95] [237]	Hardware flaws or insecure wireless communication
Key and/or Certificate Replication attack	V2I/V2V	[154] [134]	Weak certification methods and vulnerable wireless communication channel
Illusion attack	V2I/V2V	[269] [150] [294]	Insecure wireless communication
GPS spoofing/position faking attack	V2V	[269] [104] [209]	Vulnerable wireless communication
Timing attack	V2V	[291]	Non-encrypted messages, Insecure wireless communication

4.4.5.3/ ATTACKS ON INTEGRITY AND DATA TRUST:

The aim of integrity services is to make sure that any exchanged message has not been altered during transmission among the intermediate nodes. Additionally, integrity services immunize the system against destruction, unauthorized alteration or creation. External integrity attacks are not possible since a prior authentication process is required. In fact, this kind of attack is internal and integrity attacks mainly target V2V communications and not V2I communications because of the latter's fragility. Several possible methods exist and can breach the integrity property which will consequently make any transportation system defective [219]. Several examples of integrity attacks are briefly described in the following along with their possible solutions.

- **Masquerading:** Masquerade attacks are ranked second on the top five lists of electronic crimes perpetrated after viruses, worms or other malicious code attacks. The attacker seems to be an authentic user since he/she uses a valid identity which is known as a mask. This is done by forming a Black-hole or generating false messages which are then broadcast to the neighboring vehicles. This attack has different objectives such as slowing down the speed of a vehicle, changing lanes which may lead to an accident.
  -Layers targeted: Network Layer
  -Solution: To avoid this kind of attack, a Certificate-Revocation-List (CRL) is used containing the identity of detected malicious vehicles. Therefore, when malicious vehicles act in a malevolent way, their corresponding identities are distributed to the overall nodes within the network, and the CRLs are updated by introducing the identity of the new malicious cars into the list. This can reduce the effect of this attack [154], but also an efficient detection technique of malicious node is required to answer the constraints of ITS. In [395], they proposed using a combination key

Table 4.5: Different types of integrity attacks with their corresponding solutions.

Name of Attack	Communication Types	Proposed Solutions	Possible Reason
Masquerading	V2V	[154] [395]	Insecure Communication channel
Replay Attack	V2V	[285]	Vulnerable wireless communication channel
Message Tampering-Suppression-Fabrication-Alteration	V2V	[272] [123] [101] [194] [98]	Vulnerable wireless communication channel
Incorrect Data Injecting Attack	V2V	[285]	Non-encrypted message, Insecure wireless communication
Man in the middle attack	V2V	[314] [291]	Non-encrypted message, Insecure wireless communication, Poor authentication scheme

instead of a public key so that the throughput value is improved by 40%.

- **Replay Attack:** The adversary replays the valid messages sent sometime before in order to disturb the traffic. The mechanism of a replay attack consists of broadcasting a previously transmitted message [122] to ensure the objective of the transmitted message at the moment such as manipulating the location and the nodes routing tables. Therefore, this leads to mystifying the authorities and to preventing the node from knowing the sender's identity [285].
  -Layers targeted: Data Link, Network, Transport, Application Layers
  -Solution: A solution is presented which uses the cache of station (RSU or vehicles), and consists in comparing the recently received messages with new incoming messages to reject the received duplicate messages. Hence, it protects the node from replaying an attack, and makes this threat inefficient. In addition, another solution is presented which is "time stamping" for each transmitted packet to prevent the replay attacks [285].

- **Message Tampering-Suppression-Fabrication-Alteration:** The attacker here aims to break the integrity of the exchanged messages which is done by altering, removing, or creating other messages [280]. Availability and non-repudiation services are also affected. This happens when the attacker manipulates the received messages for his/her own goals. Therefore, this will lead drivers to change their decisions and for example to take a different road than the one they intended to use in the first place.
  -Layers targeted: Network Layer
  -Solution: One of these security methods is using vehicular PKI (VPKI) or

a zero-knowledge to authenticate the vehicles and to sign warning messages [123], [101], [194]. Furthermore, a group of communication can be established which is also considered an efficient method as indicated in [272]. The keys can be conducted by a Group Key Management system (GKM) [98]. In other words, if an intruder tries to attack, he/she will not be able to communicate through this closed group.

- **Incorrect Data Injecting Attack:** This kind of attack is generated from a legitimate node. Thus, this can cause hazardous effects in the network and may lead to fatal accidents [285], by creating a false message and broadcasting it or removing the traffic warning. The strategy of this attack is to hide the real safety messages from allowed users and then inject false security messages in the network.
  -Layers targeted: Network Layer
  -Solution: To defend this attack, the broadcast message should be signed and included in the transferred message. However, a non-repudiation method is necessary to reveal the attacker's identity that should be appended in the RLCs [285].

- **Man in the middle attack:** The Man in the middle Attack (MiMA) is a common attack on the communication that takes place among users. The attacker is usually situated between a minimum of two persons. The attacker here is a vehicle inserted between two communicating nodes (vehicles). The man in the middle, attacker, has the ability to control the communication between these legitimated nodes [265], so that they assume that they are directly communicating with each other. In this case, the attacker breaches the authentication, integrity and non-repudiation mechanisms.

  -Layers targeted: Network Layer
  -Solution: Using digital certificates, secure communication and good cryptography will be a good solution [314]. In addition, using an efficient authentication scheme as proposed in [291] can be another solution. It is proposed that a decentralized lightweight authentication scheme called trust-extended authentication mechanism (TEAM) can be used for vehicle-to-vehicle communication networks.
  All integrity attacks and their solutions are summarized in Table 4.5.

4.4.5.4/ CONFIDENTIALITY ATTACKS:

Confidentiality is the ability to conceal messages from a passive attacker so that any message exchanged through the network remains confidential. This is the most important point in security, that is to say to protect the data from being collected by unauthorized users. The message confidentiality in ITS can be employed for specific applications that require sharing sensitive information such as those used for toll payments using a V2I connection. For example, here the confidentiality becomes essential to provide a secure Internet connection by encrypting the message transmitted between vehicles and RSUs [387]. However, if there is no sensitive information in the transferred messages, then the confidentiality is not mandatory [154]. Encryption process can be deployed in symmetric or asymmetric ciphers [226]. The asymmetric class requires heavy computation complexity and resources compared to symmetric ones. Encrypting messages needs a session key, which is generated initially after mutual authentication between the RSU

and the vehicle. For each encrypted message, the MAC (Message Authentication Code) or a message authentication is attached to the encrypted message to add robustness against attacks. Several attacks can affect the network in the absence of confidential protection mechanism. Improper collection of clear information [285] affects the individuals' privacy, since the attacker is capable of gathering several information such as the location of the vehicle and its routes, etc. Unfortunately, the victim is not able to detect it since this kind of attack consists in listening to the media, which is easy to carry out. A list of passive confidential attacks is presented in the following and each one is briefly described:

- **Eavesdropping attack (EA):** Eavesdropping attack only influences the network confidentiality and will not have any impact on the network itself [387]. The aim of this attack is to illegally obtain access to confidential data. By spying on the data, the adversary could easily discover communication contents. It detects useful information, such as data location, which can be employed for tracking vehicles.
  -Layers targeted: Physical Layer
  -Solution: To provide resistance against this attack, the sensitive data that risks the driver's privacy (positioning and vehicle identification data) should be securely encrypted [24].

- **Traffic Analysis Attack (TAA):** TAA affects the user's privacy in addition to his/her confidentiality. This attack is extremely dangerous and consists in listening to the network for a communication pattern, then trying to analyze the collected data to extract as many useful information as possible. Sensor activities can potentially reveal enough information to enable an adversary to cause malicious harm the network.
  -Layers targeted: Physical Layer
  -Solution: The same proposition that enables one to provide resistance against eavesdropping can be used to resist the TAA [24], in addition to using VIPER (Vehicle to Infrastructure communication Privacy Enforcement Protocol Algorithm) for V2I communications [280]. It is resilient to traffic analysis attacks. In this solution, vehicle will send their messages directly to RSU and will not have vehicles acting as mix nodes.

- **Brute force attack:** It is a trial and error method used to obtain information such as a user password or personal identification number or to crack encrypted data or even to test network security. The attacker can use the brute force technique to break the used cryptographic key [223]. In a transportation environment where connection times are relatively short, a brute force attack is not easy to perpetrate, since it is time consuming and resource exhausting.
  -Layers targeted: Network, Transport Layer
  -Solution: This attack can be made inefficient by using a strong encryption and key generation algorithms which are unbreakable within a reasonable running time [285]. Another Brute force attack solution is proposed by Langley et al. [176]. In this context, a secure authentication method requires the use of some unique identification for vehicles concatenated with some large random value and then hashed using some hash algorithm.
  All confidentiality attacks and their solutions are summarized in Table 4.6.

Table 4.6: Different types of confidentiality attacks with their corresponding solutions.

Name of Attack	Communication Types	Proposed Solutions	Possible Reason
Eavesdropping attack (EA)	V2V/V2I	[24]	Broadcast nature of messages via wireless channels, un-encrypted communication channel
Traffic Analysis Attack (TAA)	V2V/V2I	[24] [280]	Vulnerable wireless communication channel, Data leakage on communication channel
Brute force attach	V2V/V2I	[285] [176]	Short cryptographic keys, and weak cryptographic methods.

4.4.5.5/ ATTACKS ON PRIVACY:

Ensuring user's privacy is one of the most important challenges in ITS. Preserving users' privacy is mainly related to preventing the disclosure of their real identities and location information. Drivers need to keep their private information protected such as their identity, their driving behavior, the past and present location of their vehicle [147, 227]. In order to preserve the privacy of drivers, each vehicle is loaded with a pool of certified pseudonyms obtained from a certificate authority [246]. One of the most popular attacks here is the Sybil attack since this granted pool of pseudonyms can be used to pretend that they are for different vehicles and send false messages to other vehicles (false traffic jams, or false alerts forcing others to modify their itinerary). The main goal of the authorities here is to ensure that the identities and their corresponding sensitive data are protected during communication. On the other hand, when an issue arises, the system operators and car manufacturers should interfere and this requires knowing the identity of the user. This indicates that a trade-off between privacy and security exists. Several privacy attacks are presented in [272] and [387] and are described below, then, common solutions are proposed.

- **Identity revealing**: Getting the identity of a given vehicle's owner could put his/her privacy at risk. In most cases, a vehicle's owner is also its driver, so it would simplify things to get private information about this person.
- **Tracking**: It allows to chase a vehicle during its journey and then discover the identity of the driver (relating the vehicle to place of work, home..) Therefore, even though the keys used usually do not use public relations to the true identity, MAC and IP addresses must change over time to avoid any possible identity disclosure [154]. MAC, IP addresses allocation, and used keys must be managed by new algorithms to avoid facing a large memory space dilemma.

- **Social Engineering**: Attackers illegally listen to the communication between V2V/V2I and misuse this confidential information. As described by Kevin Mitnick in [163], social engineering is as an "act of psychological manipulation which was popularized by hacker-turned-consultant". The attacker aims at pre-texting, phishing, and diversion theft, mainly. Due to the dynamic behavior of vehicles, social engineering attackers do not use these methods. Instead, there are three different scenarios in which a social attack can happen. (1) An attacker finds the busiest road in the area that usually has traffic and where it will be perfect to launch an attack. (2) An attacker chooses a traffic peak time (like lunch breaks, busy hours) to take advantage of the traffic. (3) An attacker starts collecting the personal data (user ID, location) which jeopardizes the privacy of users. Usually, the attacker studies different areas and chooses the best one to launch an attack on the network.

- **Identity/Location Tracking:** In this type of attack, an attacker may get a trace of the vehicle movements, and from the study of this trace, he/she can reveal the true identity of the vehicle and its personal information. An example of that is an employer in an organization who overhears a communication coming from the parking lot. Since he/she knows the identities of all the cars in the parking lot, he/she can simply know its arrival and departure dates. Another example is about a criminal organization that gains access to stationary communication boxes, then it extracts information to track law enforcement vehicles. Rental Car companies are using this ID and track the location of their own vehicles.

-Layers targeted: Privacy attacks usually target the Application Layer or Data link layer where identities are usually stored. Also, this can affect the physical layer when the credentials are stored in hardware modules (Trusted Platform Module - TPM) [171].
-Solutions: The existing privacy solutions are based on the architecture presented in [152] that defines the use of the pseudonyms to ensure an anonymous network. The responsibility to manage the vehicle identities (generation, distribution and revocation) is held by the certification authorities, which can be classified, based on a region. Therefore, a dense number of certification authorities (CAs) is required. In fact, vehicles need pseudonyms to preserve their privacy, but it is illogical to load each vehicle with a large number of pseudonyms and keys since this requires a large storage area. In addition, using the pseudonym more than once time will degrade the vehicle's privacy. Additionally, each station, vehicle or RSU, possesses a pair of private and public cryptographic keys and a unique identity. To obtain the real identity of the Vehicle, a judgment should be required. The limitations of this protocol is that it requires vehicles to store a large number of pseudonyms and certifications, where a revocation scheme for abolishing malicious vehicles is difficult to implement. It is preferable to preserve the location privacy of a vehicle by breaking the linkability between two locations, for which the vehicle can update its pseudonym after each transmission. Taking into account that a powerful adversary may still link the new and old pseudonyms by monitoring the temporal and spatial relations between new and old locations, the techniques of mix zone [145] and silent period [146] have been proposed to enhance the pseudonym scheme. Each vehicle in a mix zone will stay silent during transmission, and randomly update its pseudonym when it travels out of the mix zone and becomes re-activated.
In [134], they proposed to use a set of anonymous keys that can be preloaded in the vehicle's TPD (Tamper Proof Device). Each key is certified by the CA and is used for a short time, which means that it must be changed frequently.

Lu et al. [179] proposes a privacy preservation protocol (ECPP) for anonymous authentication. The protocol uses short time anonymous keys between On-Board Units (OBUs) and Roadside Units (RSUs). The anonymous key needs a minimum of storage to avoid losing the security level. The network architecture is composed of the trusted authority (TA), the immobile RSUs on the roadside and the mobile OBUs equipped on the moving vehicles.
Zhang et al. [233] introduced a novel decentralized group-authentication protocol rather than by a centralized authority, where each RSU is used to maintain and manage an on-the-fly group within its communication range. Vehicles entering the group can anonymously broadcast vehicle-to-vehicle (V2V) messages, which can be instantly verified by the vehicles in the same group (and neighboring groups). Later, if the message is found to be false, a third party can be invoked to disclose the identity of the message originator. This protocol efficiently exploits the specific features of vehicular mobility, physical road limitations, and properly distributed RSUs. If some RSUs unexpectedly collapse, only the vehicles that are driving in those collapsed areas will be affected. Due to the numerous RSUs sharing the load to maintain the system, performance does not significantly degrade when more vehicles join the VANET; hence, the system is scalable.
In [259], they propose using group signature, where one group public key is associated with multiple group private keys. In the AMOEBA [155], vehicles form groups. The messages of all group members are forwarded by the group leader, which implies that the privacy of group members is protected by jeopardizing the privacy of the group leader. In case a malicious vehicle is selected as a group leader, then, all group members' privacy may be leaked by the malicious leader.
Zhang et al. [261] proposes a vehicular authentication protocol called "APPA" to trust the vehicular communications and privacy of vehicles. This protocol is identity-based cryptography, aggregate signature and one time signature. If a vehicle obtained a secret key from a trusted authority (the secret key is associated with the vehicle's identity), it could sign messages. The signature on a message uses the vehicles identity which is a one-time pseudonym.
Although privacy has been recognized as a serious problem, robust technologies and architectures still have to be developed in order to ensure the users' privacy. Discussed privacy threats and solutions are all summarized in Table 4.7.

Table 4.7: Different types of privacy attacks with their corresponding solutions.

Name of Attack	Communication Types	Proposed Solutions	Possible Reason
Identity revealing	V2V	[134] [152] [337] [145] [146] [179] [233] [259] [155] [261]	OBU vulnerabilities, reusing pseudonyms, insecure wireless communication.
Tracking	V2V		
Social Engineering	V2V		
Identity/Location Tracking	V2V		

4.4.5.6/ ATTACKS ON NON-REPUDIATION

It is the security mechanism in which the sender/receiver can prove that a transaction occurred while preventing the receiver/sender from denying that. This prevents false denials

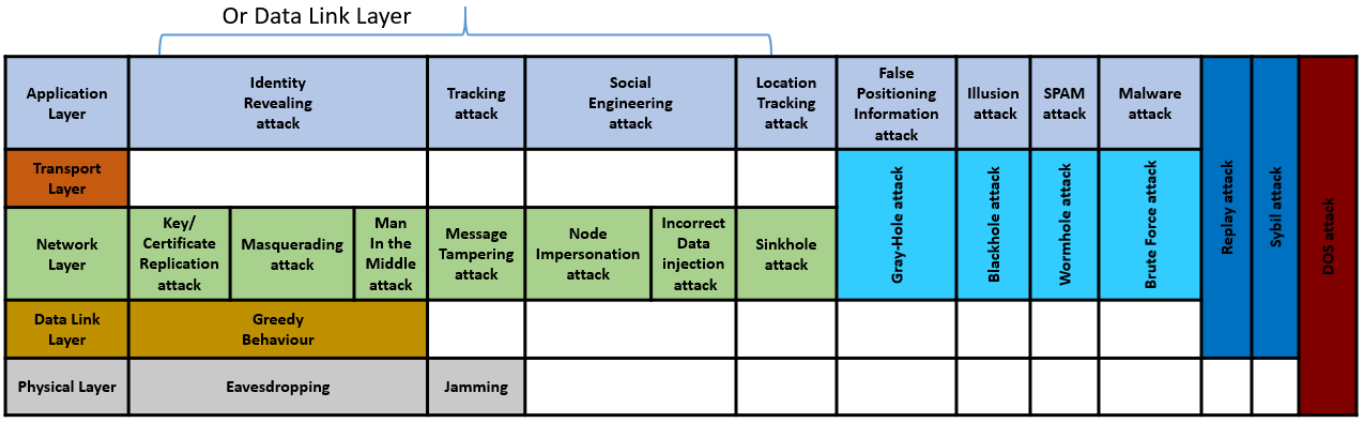


Figure 4.9: Attacks with their corresponding Internet Protocol Stack layers

involved in the communication. The main aim of non-repudiation consists in collecting, maintaining, making available and validating undeniable evidence about a claimed event or an action in order to resolve disputes about the occurrence or non-occurrence of that event/action. Non-repudiation depends on authentication, but it generates an evidence in the system that can identify the attackers who will not be able to deny their crimes [315]. Any car information will be saved in a Tamper Proof Device (TPD) and any authorized official will be able to retrieve these data.

-Layers targeted: Application Layer

-Solutions: Three different solutions exist for non-repudiation: (1) Public Key Infrastructure [182] (2) ID-Based crypto-system [128] [178] and (3) Situation Modeling-Based mechanism. In PKI solution data authentication and non-repudiation can be performed using the digital signature, which is implemented using asymmetric cryptography where each entity has two keys: a public key and a private key. The ID-Based crypto-system uses any known information which represents the identity of the user for the purpose of verifying the digital signature. This public information could be an email address, network address, user name or any combination of these identities. The third solution, situation Modeling-Based mechanisms, is based on generating VANET/IoV models for vehicle driving trends and routines to enhance the security by nodes management [299, 158]. The challenges in these non-repudiation solutions are presented in Table 4.8.

Finally, at the top level, all the aforementioned attacks can be classified according to the layer(s) they affect, in the network protocol stacks as shown in Figure 4.9. Each layer is presented as a specific color, and the attacks having the same color means that they occur upon this specific layer. Moreover, attacks like Gray-Hole, Black-Hole, Wormhole,Brute force, Replay, Sybil and DoS are represented with different colors since they affect more than one layer. Finally, as seen at the application layer some attacks can occur on data link as well.

Table 4.8: Non-repudation challenges in different architectures.

VANET ARCHITECURE	Challenge
PKI	Need communication resources and large communication overhead.
Identity based	Risking the ID- privacy of the Vanet users.
Situation Modeling	Complexity in different proposals.

4.4.6/ AN ITS RISK ANALYSIS STUDY

ITS are cooperative systems based on vehicular communications and are considered as a compromising approach to enhance road safety, efficiency and convenience. VANET and IoV pose several research challenges, especially on the aspect of security, since various elements are used such as communication architecture, applications, and protocols. Moreover, the existing literature focuses on preventive attack techniques to achieve security protection. In this section, a simple risk assessment is presented, while less work in this field is presented. However, a new risk assessment method is required to quantify the security risk of ITS attacks, which is a complicated task. Addressing threats in ITS and analyzing their associated risks is an initial step to define new security solutions adapted to ITS applications and communications. Previously, some challenges and threats have been described, which will help to quantify the strength of attacks when necessary. The proposed risk analysis based on ETSI Threat, Risk, Vulnerability Analysis (TVRA) methodology [278] is based on the product of the likelihood of an attack and the impact of the attack on the system. The system assets and its associated threats in addition to the threat agent that tries to break the system should be identified by the TVRA method. Therefore, the outputs of TVRA are a measure of the risk of identified threats and can be determined based on their estimated value of likelihood and impact upon the system. In the following, several countermeasures and security frameworks are specified taking into account both ITS application constraints and the developed risk analysis. Three levels of risk are defined: Minor, Major, and Critical. Threats ranking as Critical mean that an urgent and priority countermeasure should be defined, while Major risk should also be treated with a lot of attention. On the other hand, threats that possess minor risk get less attention in the study. The existing threats that can be ranked as critical and major are represented in Table 4.9 risk analysis.

Table 4.9: Qualitative risk analysis according to [278]

Kind	Threat	Needed Attacker capabilities	Motivation of the attacker	Likelihood	Impact	Risk
Availability	Flooding/ Spamming	No rating	Moderate	Possible	Medium	Major
	Black hole	Moderate	High	Likely	High	Critical
	Malware	Basic	High	Likely	High	Critical
	Jamming	Basic	Moderate	Possible	Medium	Major
	RF Fingerprinting	Extensive	Low	Unlikely	Low	Minor
Authentication	Masquerade	Moderate	Moderate	Possible	High	Critical
	Sybil attack	Extensive	High	Possible	High	Critical
	Illusion attack	Extensive	High	Possible	High	Critical
	GPS Spoofing	Moderate	Moderate	Possible	High	Critical
	Sensor spoofing	Extensive	Moderate	Unlikely	High	Major
	Replay	No rating	Low	Possible	Medium	Major
Integrity	Manipulation of messages	Moderate	Moderate	Possible	Medium	Major
	Injection of false message	Moderate	Moderate	Possible	Medium	Major
Privacy &Confidentiality	Eavesdropping +data analysis	Extensive	High	Possible	Medium	Major
Privacy	Location tracking	Basic	High	Likely	Medium	Critical

4.4.7/ MODERN SECURITY LAYERS

The security layer in ITS in Figure 4.10 is composed of 4 sub-layers:

- **Material layer**: Several physical resources are used to reach the objectives of ITS such as OBU, GPS, radars, Event Data Recorder (EDR), antennas, etc. They should be protected in order to resist physical attacks. For that reason, these devices could be secured and should be built according to the Trusted Platform Module (TPM) specifications [131]. Furthermore, TPM is a hardware piece that can protect and store data in shielded locations [171] by employing a software infrastructure.

- **Authentication layer**: The authentication layer should ensure all kinds of authentications which are users, source and location authentication. The users' authentication prevents unauthorized users to access the system. Moreover, source authentication permits receivers to verify the source entities and ensure data integrity [206]. For that reason, the digital signature is used and requires the existence of the vehicular PKI. In the context of transportation systems, the location authentication is necessary and permits the receiver to verify the sender's position. These authentications are conducted progressively and not simultaneously.

- **Trust layer**: The implementation of the trust layer consists of two parts, where the first one is a trust system [253], and the second one can be a reputation [117] or a Plausibility Check System(PCS) [215]. The importance of the trust layer is the validation of communication and it can provide the non-repudiation requirement. Indeed, the trust system consists in implementing the TPM mechanisms. The reputation system builds an opinion concerning a node that wishes to communicate. This opinion is obtained by analyzing several collected local information such as speed, position, acceleration, etc., as well as information from other internal users. On the other hand, if a PCS is used, a verification process is required to ensure that this information corresponds to this specific event. As a conclusion, the main goal of the trust layer is to discover which nodes are trustworthy. An important requirement that can help the trust layer is to collect sufficient information about the sender's node, which permits to register these traces in order to support their corresponding consequences. Let us say that the trust layer should provide resistance against availability attacks.

- **Privacy & Data confidentiality layer**: The main goal of this layer is to preserve the privacy of users and their sensitive information in the network. Several solutions have been described previously to reach this objective. For that reason, sensitive information should be encrypted before being sent to avoid attacks on this layer.

It can be noted that the cryptographic primitives are implemented at both Authentication Layer and at the Privacy and Confidentiality Layer.
This approach could be illustrated to ensure three major objectives:

- **Prevention:** Resembled by Security Material Level and the Authentication Layer.

- **Detection:** Resembled by the Trust level.

- **Privacy:** Resembled the Privacy and Confidentiality Layer where cryptographic primitives are mainly implemented.

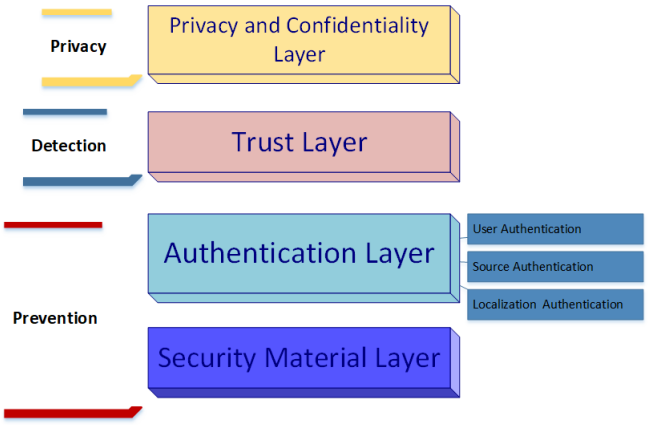


Figure 4.10: Security modern layers and their corresponding objectives.

Table 4.10: Required security services for VOB, IOB and IUV

Security Service	VOB	IOB	IUV	Mechanism
Confidentiality	✗	✗	✓	Encrypting only sensitive messages; Randomizing Traffic Patterns
Authenticity	✓	✓	✓	Message signature Trusted Hardware Module; Active Detection Systems
Integrity	✓	✓	✓	Message signature and other integrity metrics for content delivery
Non-repudiation of source	✓	✓	✓	Message signature
Authorization and privilege classes	✓	✓	✓	Certificate accompanying message signature
Non-repudiation of receipt	✗	✗	✗	Not mandatory
Anti-replay	✓	✓	✓	Message signature containing verifiable time variant data
Plausibility verification	✓	✓	✓	Check mechanisms ensured by IEEE P1609.2.
Availability	✓	✓	✓	Pseudo-random Frequency Hopping Access Control and signature-based authentication
Privacy protection measures	✓	✓	✓	Pseudonymity, Unlinkability ID-based/PKI based System for User Privacy

4.4.8/ SECURITY SERVICES FOR COMMUNICATION TYPES IN ITS

According to [390], three kinds of communication are presented, which are Vehicle-Originating Broadcast (VOB), Infrastructure-Originating Broadcast (IOB), Infrastructure-Vehicle Unicast (IVU). In Table 4.10 the required security services are described for each communication type. The application scenario defines the security services required to ensure a safe implementation. The three kinds of communication are described briefly in the following:

- **Vehicle-Originating Broadcast (VOB)**: The vehicle is the origin of the broadcast exchanged message that contains information about the behavior of the source such as its movements and safety (in ETSI CAM message is employed). This will inform the neighboring vehicles and will reduce dangerous hazardous situations. VOB is essential for road safety applications.
- **Infrastructure-Originating Broadcast (IOB)**: IOB communication is used by all vehicles under the communication range of a specific RSU. It will inform the vehicles, which are in the vicinity of a specific road infrastructure location, by safety and mobility information. This will ensure a better safety for data which is relevant to all vehicles. Indeed, the security service requirements for IOB are similar to that of VOB with few differences [262].

- **Infrastructure-Vehicle Unicast (IVU)**: IVU communication is employed for unicast transactions between a vehicle and RSU or vice versa. In fact, IVU can be used for commercial and comfort applications. The required security services for the different communications VOB, IOB, IUV are shown in Table 4.10. From a security viewpoint, the non-broadcast nature of IVU requires ensuring confidentiality in addition to the security service requirements of VOB and IOB.

4.5/ EXISTING SECURITY ARCHITECTURE FOR ITS

Current security-system/architecture-based researches can be classified into the following three categories with reference to different technology vantage points. Current researches classify the security system architecture into four different cryptographic categories: (1) **Public Key Infrastructure (PKI) based** schemes (PKI based approaches, pseudonym-based approaches and group signature-based approaches), (2) **Crypto-Based security based** schemes (3) **Non-fully PKI based** schemes (identity based cryptography and hybrid approaches etc.), and (4) **Situation modelling-based** security system architecture which will be discussed briefly in the following.

4.5.1/ PKI-BASED SECURITY SYSTEM ARCHITECTURE

PKI is used for the asymmetrical algorithm based security applications. In addition, PKI provides several security services such as certificate generation, renewal, and cancellation, signing and issuing, check, maintenance, audit, etc. The certificate is provided by the PKI link public key in the public/private key array with the owner's identification and encryption technology. PKI requires using CRL (Certificate Revocation List) in order to ensure a safe and secure management in real network implementation. This requirement can be considered as a critical problem and introduces high communication overhead. Currently, a list of recent security schemes that are based on PKI and a comparison of the communication overhead is presented in [312]. In [230], a TPM-based security architecture is proposed which can form trusted grouping through PKI security mechanism. Its robustness is proven in [231].

4.5.2/ CRYPTO-BASED SECURITY

In [188], the authors provide a new security scheme for ITS that can provide privacy, data confidentiality and integrity, and non-repudiation by using a symmetric block cipher algorithm and a certificate-based public key cryptography scheme. The privacy and data confidentiality is ensured by employing the robust block cipher AES [386]. On the other hand, to provide data integrity, source authentication, and non-repudiation, the exchanged message is signed by using the sender's private key. This scheme is based on certificate-based public key cryptography so an overhead of such a scheme can create a delay in transmission. This scheme suffers from the latency limitations and can be considered as non-suitable for time-critical safety applications.

4.5.2.1/ ANONYMOUS AUTHENTICATION PROTOCOL

An anonymous authentication protocol is proposed in [282] for V2I communication and is based on Certificate-based Cryptography (CBC). This proposition can ensure conditional privacy and non-repudiation.

4.5.2.2/ MESSAGE LINKABLE GROUP SIGNATURE (MLGS)

Qianhong Wu et al. [232] proposed a new privacy-preserving technique called Message Linkable Group Signature (MLGS). This technique provides anonymous authentication. In this technique, it is assumed that most of the vehicles in the network are honest. In this system, a threshold mechanism is used as a priori countermeasure. A message is considered as trustworthy if at least n vehicles endorse this message. This n is a threshold which is adaptive. The sender can change the threshold. If a node produces two signatures on one message then a trusted authority will identify it as an attacker. A Sybil attack can be avoided by using this technique. If a vehicle receives a message with multiple signatures, it can check whether these multiple signatures are from a single vehicle or from multiple honest vehicles.

4.5.3/ ID-BASED SECURITY SYSTEM ARCHITECTURE

To cancel the overhead of CRL and to avoid the use of PKI, the architecture of ID-based security system is presented. Moreover, an ID-based Encryption algorithm is used for the generation of a pseudonym, which is renewed as required. It is feasible for an entity to possess assemble groups made of several pseudonyms for privacy defense when the authentication and the signature are approved. ID-based security method must guarantee ID privacy, a precondition for protection of user's safety and privacy. The key point lies in generating an irreversible algorithm for pseudonyms based on ID with the firm confirmation that only one pseudonym is available within the same entity to prevent a Sybil attack. Indeed, in [393], they employ ID-based encryption for pseudonym generation and conduct the control of signatures and identify authentications through a threshold scheme to satisfy security and privacy requirements. In addition, a method for trust domain division pursuant to common domain in use is presented. Additionally, in [396], they combine the ID-based signature (IBS) scheme with the ID-based online/offline signature (IBOOS) scheme apart from ECC based short digital signature for time-validity improvement.

4.5.3.1/ ID-BASED AUTHENTICATION SCHEME

An authentication framework for RSUs and Vehicles is presented in [274] and it uses ID-based encryption. The process of authentication consists of three kinds of authentication, which are V2I authentication, V2V authentication, and I2V authentication. The authentication between the RSU and the vehicle is ensured by using an ID-Based Signature (IBS) scheme, while the authentication between vehicles is guaranteed by employing ID-Based Online/Offline Signature (IBOOS) scheme. The registration of vehicles at Regional Trusted Authority (RTA) is an initial and important process which enables a vehicle to drive on a road. After that, RTA generates corresponding certified domain parameters in response to an authentication request, and publishes it. Then, this certification is hashed

Table 4.11: Security Solutions for different features

Taxonomy	Entity-base Rep.(VARS) [117]	Data-Centric Trust-Model [183]	Event-base Reputation [198]	ID-based Auth [274]	MLGS [232]	RaBTM [284]	ID-Based Security [393]	D. Sig. &Pwd [396]	Cert. based Auth [282]	EDR [208]
Authentication				✓	✓		✓	✓	✓	
Non-Repudiation				✓			✓		✓	
Integrity		✓						✓		
Privacy				✓	✓	✓	✓		✓	✓
Confidentiality							✓			
Privacy (pseudonymity)				✓	✓	✓		✓	✓	
Reputation	✓	✓	✓			✓				
Revocation										✓

and stores its corresponding hash values in a database instead of its real ID. In addition, the vehicle privacy is ensured by using the pseudonyms, which are self-generated identifiers. A vehicle changing its pseudonym should unicast its new generated one to RSU. After that, RSU verifies the received pseudonym by checking the signature and accepts if its authenticated. However, this scheme suffers from latency limitations and can be considered as unsuitable for safety critical applications, since safety application cannot support a delay of even milliseconds, especially when the pseudonym of a new vehicle is not known by the other vehicles.

4.5.3.2/ IDENTITY-BASED ENCRYPTION SCHEME

A secure identity based cryptography scheme is presented in [229] that generates the public key from a public unique identity of station. As a consequence, the overhead that is added when a certificate scheme (CRL) is used is reduced. The traceability and privacy are provided by using a pseudonym, which can be generated by RSU or the vehicles. The privacy preservation and non-frame ability against misbehaving nodes are achieved by using threshold signature and authentications. An important issue of this scheme is the revocation of a user's public key, since when a user's public key is revoked, that means that the corresponding identity is changed, which is inconvenient.

4.5.3.3/ PAIRING-BASED DECENTRALIZED REVOCATION

A revocation protocol is proposed based on pairing Efficient Decentralized Revocation (EDR) [208]. This protocol is based on probabilistic random key distribution and its nature is decentralized and permits to build a group of legitimate neighboring vehicles, which can revoke a nearby malicious vehicle by applying a vote and the result will exceed the threshold.

4.5.4/ SITUATION MODELLING-BASED SECURITY SYSTEM ARCHITECTURE

According to [154], an architecture of flexible secret key management and trust information was presented and called SAT (situation-aware trust). According to specified situations, SAT is able to build up compatible trust mechanisms. SAT also discusses a rapid establishment of trust mechanism by means of popular social websites. In [134], the authors assume that the vehicles on the road of location act as intelligent agents, based

on which trust models are built. Several reputation systems are presented such as Vehicle ad-hoc Reputation System (VARS) [117], Data-Centric Trust Based Security [183], Event-Based Reputation System [198], and Trust Management System - RaBTM [284]. Finally, the solutions for the security features are all presented in Table 4.11.

4.6/ CONCLUSION

VANET/IoV are both interesting fields of modern emerging networks and their importance comes from their practical benefits related to the safety of human lives, especially for ITS applications such as road safety and traffic management. Therefore, securing ITS applications is a great challenge since these applications suffer from several limitations. In this chapter, ITS applications, characteristics, recent standardization works, threats and their impact on the system were all discussed. Also, the evolution of VANET to IoV and their differences are stated. The main difference however is the heterogeneity introduced in IoV which adds more challenges for researchers. This chapter identifies all existing security issues and challenges and then classifies them from a security viewpoint. The attacks were classified according to their impact they cause on the security service they target and also they are classified according to the network layer they affect. The solutions for every attack that were proposed in the research works are also stated. Then, we also list some of the existing security architectures that are used in Intelligent Transportation Systems. To sum up, this chapter will help researchers get a clear and a detailed look at every aspect of security issues in VANET/IoV and will pave the way for them to innovate and find new practical solutions.

5

TestU01 and Practrand: A Randomness Evaluation for Famous Ciphers

"Random numbers should not be generated with a method chosen at random" — Donald Knuth

5.1/ Introduction

After investigating one of the most challenging platforms, we start looking deeper into methods that tackle the problem of proposing new successful cryptographic solutions in such systems or any other system. In today's technological revolution, a security guarantee has become a major issue and a basic need for users, companies, applications, and researches alike. Many new terms and technologies have invaded the industry and the research fields. Internet of Things(IoT) is one of the most promising research topics in both engineering field and business. Connected devices are increasing day after day. In fact, Cisco's Internet of Things Group (IoTG) estimated the number of connected devices to reach 50 billion by year 2020 [360]. These connected devices exchange different kind of information as streaming of stored multimedia content(audio, video), live streaming (video conferencing, online gaming), and real-time interactive multimedia communication such as the case of surveillance. These exchanged packets should be transmitted and received continuously. However, these packets of data need to be secured against all the powerful new attacks. In order to go along with this tremendous generation of data, cryptographers are excelling at finding new solutions that can respect the new requirements. As such, using Lightweight Cryptography (LWC) has become one of the foremost desired solutions in security especially for limited sensors. It investigates the implementation of cryptographic algorithms for resource constrained devices [385, 368] that are excessively used in today's networks. The use of limited resources, batteries, sensors, and Wireless Sensor Networks justifies the need for efficient lightweight cryptography. Smaller block sizes, modest key size, little code measure, fewer clock cycles and lower number of rounds are all factors that can cause any cipher to become lightweight. However, there must be a trade-off between the security of the cipher and the limitations of the constrained devices. In order to have an efficient and reliable cipher, it must undergo a certain number of tests. One of these tests is the **randomness test**. The security provided by these cryptographic algorithms is directly related to randomness, since compromising the

randomness of the ciphered output will jeopardize the whole system. Ignoring this criterion would be extremely dangerous, since the recovery from a security breach can be extremely expensive. **Randomness can be defined as the outcome of a probabilistic process that produces independent, uniformly distributed and unpredictable values that cannot be reliably reproduced [156]**. The main concern in randomness is being able to produce an unpredictable output, which is an uncorrelated output having a uniform distribution with the lack of bias. Random outputs must be unpredictable, irreproducible and should prevent any attacker to learn/predict former or subsequent values. Yet, in the absence of qualified randomness tests, the quality of the cipher will not be verified. However, random sequences can generally have specific statistical properties that can be measured using different statistical tools. Some of these tools are TestU01 [148], Practrand [216], DieHard [58] and ENT [187] etc... These tools try to avoid sequences which do not verify certain statistical properties, but cannot guarantee perfect randomness. However, passing these tools will grant the cipher the minimum required randomness validity and will highly assure the statistical properties it must possess. In this work, several crytographic algorithms were implemented using Practrand and TestU01, since these tools, even if they can be described as simple to use, validate the randomness of the cipher text and are considered to be efficient. Some of the algorithms implemented failed these statistical tests which shows the importance of implementing any new proposed cryptographic algorithm into these tools. This work presents a practical proposal for cryptographers to validate the randomness produced by their algorithms. It can guarantee the randomness level desired in newly proposed ciphers that are responsible for protecting the different kinds of data exchanged.

5.1.1/ IMPORTANCE OF RANDOMNESS

Having this amount of exchanged contents, producing an output that can be random enough to prevent any recovery of data is the main concern of any cryptographer. In order to increase the randomness produced by any cipher, one of the most important elements of the cipher is the **Cryptographic Key**. It holds the security of the whole system, so, the generation, agreement, storage and destruction of the key must be well managed. Any information about the key will lead to the knowledge of the secret message and thus a security breach will occur. Cryptographic keys must be long enough, must have a large key space and must be generated by a complex and efficient method. The keys must be unpredictable, thus must have high uncertainty (high entropy), highly independent bits, a uniform distribution, and cannot be reproduced, thus, in other words, the keys must be random. Using **Initialization Vectors** also adds randomness to the system where the same key can generate different unique outputs by adding an initial vector into the process. **Cryptographic Salts** can also add randomness especially when used for passwords to avoid easy carried out dictionary attacks [113]. **Padding strings** are also implemented extensively in key block ciphers to avoid compromising short messages and to disguise the original length of the message by adding a random padding to the plain-text block. **Nonces**, numbers used only once, are also of great importance in cryptography to avoid reusing any value [108].
As can be seen, many efforts have been exerted to add randomness to the cipher to prevent different possible attacks. An insufficient degree of randomness will expose the system to differential attacks [35], chosen-plain text attacks, known-plain text attacks etc... Therefore, randomness tests are the least cryptographers can do. **The main contribu-**

tion of this chapter is to accentuate the importance of using Practrand and TestU01 that can eliminate to a great extent any doubts in the randomness of the ciphered output. The aim is to show that any new proposed cipher should at least pass the tests simulated by these available, simple and free tools.

5.2/ BACK GROUND AND AN OVERVIEW

In this section, the algorithms tested during this proposal are presented. After that, an explanation for TestU01 and Practrand will be added to create a better sight for these tools.

5.2.1/ ALGORITHMS IMPLEMENTED

In order to validate the given proposal, a set of cryptographic algorithms were tested. These algorithms vary in their key space, number of rounds and mode of operation. They were selected in this work because they are well-known for their good security measures or are new proposals that need to be securely validated. They are listed below and are represented in Table 5.1:

- PRESENT [141]: Present is considered as one of the lightweight and ultra-lightweight ciphers. In fact, it represents a milestone in the field of lightweight cryptography. It uses 80-128 bit key with 64 bit blocks through 31 rounds. PRESENT is one of the first ciphers implemented on ultra-constrained devices.
- IDEA (International Data Encryption Algorithm) [23]: In order to reduce the memory overhead, IDEA uses only XOR, addition and modular multiplication operations. It uses 128-bit key with 64-bit blocks through 8.5 rounds where all data operations are performed in 16-bit unsigned integers.
- LBlock [257]: LBlock is a lightweight algorithm. It uses 80-bit keys and 64-bit blocks through 32 rounds. The authors chose to apply diffusion on half of the data in each round and a simple rotation on the other half, therefore, it produces ultra-lightweight implementations in both hardware and software.
- HIGHT [127]: It is another lightweight cipher based on simple computations and operations. It uses a robust round function avoiding the use of S-boxes where the key is 128-bits and 64-bit blocks are processed through 32 rounds.
- TEA (Tiny Encryption Algorithm) [38, 140]: TEA is a block cipher known for its simplicity in implementation on both hardware and software. It operates on two 32-bit unsigned integers that can be derived from a 64-bit data block, and uses a 128-bit key. It has a Feistel structure with a suggested 64 rounds.
- XXTEA (Corrected Block TEA) [61]: XXTEA was proposed as a correction for TEA algorithm that suffered from several weaknesses [90]. The block has an arbitrary size, at least two words (64 bits), and the key size is 128 bits where the number of round is dependent on the block size. The number of full cycles to perform over the block is given as $6 + 52/n$ (6-32 full cycles) where n is the number of words in the block.

- RC4 [63]: The famous RC4 (Ron's Code) generates a pseudo random stream of bits (a keystream). These streams can be used for encryption by combining the generated stream with the plain-text using bit-wise exclusive-or. The most often used key is 16 bytes (128 bits), but other keys used can be between 40-bits and 256-bits.

- RC4D [370]: RC4D is an enhancement for RC4 proposed by Michael Kwasnicki. The author added a level of diffusion into the original RC4 and proposed four different versions of his code which are: **k,i,p,e**.
  Implementations with a **k** have a fixed key length of 16 bytes (128 bits) which leads to a more than $2\times$ increase in speed and a decent size reduction. Implementations with an **i** perform the encryption in-place. It will reduce the need for intermediate buffers and will also reduce the pressure on the scarce RAM during en/decryption while providing a small increase in speed and a small reduction in size. Implementations with a **p** perform a pre-computation of the S-Box using the key. Due to the aimed small plain-text size, the S-Box is very large compared to it and also the computation thereof. A pre-computed S-Box reduces this overhead for every en/decryption to just one call of memcpy. This provides a $3\times$ speedup but comes at the expense of RAM. Moving this to EEPROM proves to be ineffective as can be seen at the implementation with an **e**.

- ChaCha [165]: It is a modification of Salsa20. ChaCha is a stream cipher which uses a 256-bit key and 64-bit Nonce and is based on the 8-round cipher Salsa20/8. The changes made are designed to improve diffusion per round, thus increasing the resistance to cryptanalysis, while preserving and improving time per round. Round number in ChaCha can be 8, 12 and 20 as well as 128 and 256 bit keys. In this work, ChaCha20 was implemented with a 128 bit key.

- Blowfish [37]: It is a symmetric-key block cipher with a 64-bit block size and a variable key length from 32 bits up to 448 bits. Blowfish is a 16-round Feistel cipher and uses large key-dependent S-boxes and a highly complex key schedule.

- Twofish [59]: It is a symmetric-key block cipher derived from Blowfish cipher, with a block size of 128 bits and key sizes 128, 192, 256 bits. The number of rounds is 16 and it has a Feistel network structure.

- 3DES [40]: It is a symmetric-key block cipher, which applies the DES cipher algorithm three times to each data block. It was invented since the original DES turned out to be weak and easy to break. 3DES uses 48 rounds in its computation (transpositions and substitutions), and has a key length of 168, 112 or 56 bits.

- HC-128 [190]: HC-128 is known to be a simple and secure stream cipher. From a 128-bit key and a 128-bit initialization vector, HC-128 generates a keystream with length up to 2^{64} bits. HC-128 was designed to prove that a strong stream cipher can be built from nonlinear feedback function and nonlinear output function.

- Camellia [69]: Camellia is a block cipher with symmetric key operating in either 18 rounds for a 128-bit key or 24 rounds for 192- or 256-bit keys, and has a block size of 128 bits. Camellia was designed to be efficient for both software and hardware implementations and it is used in various devices from low-cost smart cards to high-speed network protocols.

- Rabbit [91]: Rabbit is a lightweight stream cipher, famous for its high-speed. It creates a key stream from a 128-bit key, a 64-bit initialization vector, and 513 bits of internal data. The cipher was designed with high performance in software in mind.

Table 5.1: Cryptographic algorithms tested by TestU01 and Practrand.

Block Ciphers			
Algorithms	Key Length (bits)	Block size (bits)	Round number
Hight	128	64	32
Camellia	128 192,256	128	18 24
Lblock	80	64	32
Present	80 128	64	31
TEA	128	64	64
XXTEA	128	>64	>6 <32
BlowFish	>32 <448	64	16
TwoFish	128, 192, 256 bits	128	16
IDEA	128	64	8.5
3DES	168, 112, 56	64	48
Stream Ciphers			
Rabbit	128	1	1
RC4	40-256	1	1
RC4Dkip	40-256	1	1
ChaCha	128 or 256	1	8-12-20
HC-128	128	1	1

5.2.2/ AN OVERVIEW OVER TESTU01 & PRACTRAND

As explained earlier, randomness is important in cryptography to ensure: (1) randomness of the cryptographic keys, (2) security against attacks, (3) privacy and anonymity, (4) and to ensure unpredictability. This can only be achieved by using high quality randomness validation tools. The two tools that are used in this work are TestU01 and Practrand that operate distinctly. TestU01 is a comprehensive C library that contains examples of PRNGs, utilities and a collection of statistical tests drawn from the academic literature of RNGs, whereas Practrand is a C++ library of pseudo-random number generators (PRNGs, or just RNGs) and statistical tests for RNGs. Previously, PractRand (standard, 1 terabyte) found bias in 78 PRNGs while TestU01 (the BigCrush) found bias in 50 PRNGs. Each tool has its own means to define the level of randomness. For example, Practrand is the only test suite to allow functionally unlimited test lengths. It requires more bits to find bias than any other test suite, and multi-threading is supported, but maximum speedup tends to be limited to about 3x. While for TestU01, there exist three tests: SmallCrush, Crush, and BigCrush. TestU01 is the only test suite with a big academic name behind it, and the only test to guarantee (on default settings anyway) that all subtest results are

100% independent. However, it does not support multi-threading. Below is an overview on how Practrand or TestU01 operate:
Tests in PractRand:

- BCFN: checks for long range linear correlations (bit counting); in practice this usually detects Fibonacci style RNGs that rely upon large lags to defeat other statistical tests. Two integer parameters are used:(1) the minimum ”level” it checks for bias at (it checks all higher levels that it has enough data for), higher values are faster but can miss shorter range correlations. The recommended minimum level is 2, since that helps it skip the slowest parts and avoids redundancy with DC6 checking for the shortest range linear correlations, while still doing a reasonable amount of work considering how much memory it has to scan. (2) The second integer parameter helps to determine the amount of memory and cache it will use in testing. It is the log-base-2 of the size of the tables it uses internally. Recommended values are 10 to 15, larger values should be used if cache is large, lower values should be used if cache is small. Each individual ”level” of this is a frequency test on overlapping sets of hamming weights.
- DC6: checks for short range linear correlations (bit counting); takes several parameters that determine the size of the integers it operates on internally, the number of adjacent such integers it looks for correlations between, and which information it uses for each such integer; it is a frequency test on overlapping sets of hamming weights.
- Gap16: A variation on the classic ”Gap” test. Usually, the gap test is used to determine the significance of the interval between recurrence of the same digit.
- BRank: A standard binary matrix rank test. The most original part of it is the control logic that decides when data should be taken from the RNG output stream to make a matrix and what size matrix it should be. The parameter is a log-scale amount of time per gigabyte it spends calculating matrix ranks. Due to the coarse-grained nature of the results it produces, precise p-values are impossible for many of its subtests.
- FPF: ”floating point frequency” test; it is purely an integer math test. This checks for very short range correlations, even shorter than DC6, especially those correlations involving lots of 0 bits. Technically speaking, this test does a frequency test applied to the binary format of floating point numbers storing the integer values of overlapping windows of the original data stream.

Tests in TestU01:

TestU01 is implemented in the ANSI C language, and offers a collection of utilities for the statistical testing of uniform random number generators (RNG). TestU01 gives the user four groups of modules to analyze the desired RNGs: (1) Implementing pre-programmed RNGs, (2) implementing specific statistical tests, (3) implementing batteries of statistical tests, and (4) running tests to all RNGs families. The tests are applied to a sample of size n produced by the RNG. The p-value will range between 0 and 1. Tests executed by TestU01 will enable one to know the optimum sample size that should be used before the generator starts failing. There are three different battery tests in this library: The Small Crush (10 tests, around 8 seconds), Crush (96 tests, around 30 minutes) and The

Big Crush (160 tests, minimum 4 hours). The main aim for any generator is to pass the Big Crush test whose execution can take up to 24 hours to be completed and uses 238 random values. It lists the p-values and shows those who come outside the [0.01,...,0.99] interval. However, the drawback of TestU01 is that it works with a fixed amount of data and discards the least significant bit (for some tests even two bits) of the 32-bit numbers being tested. It is important to state that for academical reasons, TestU01 is designed to test 32-bit numbers, however, the random number generators used today produces 64-bits. Indeed, years ago, most random number generators would produce, at best 31-bit random values. In this work, we cast the output of 64 bit test to a 32-bit test.

5.3/ Proposed Scenario and Randomness Evaluation

In this section, all the aforementioned cryptographic algorithms are tested. All the codes are implemented using the C language. Two libraries are used to test these algorithms in addition to some codes implemented manually from trusted repositories. The two libraries chosen are Libgcrypt [120] and Wolfcrypt [351]. The wolfSSL library is known to be a lightweight, portable, C-language-based SSL/TLS library that is targeted at IoT because of its size, speed, and feature set. It works seamlessly in desktop, enterprise, and cloud environments as well. This library has been selected since there are two versions of the WolfCrypt cryptography library have been FIPS 140-2 validated. Hence, it can be advocated that it is a well known and reputed cryptographic library. The other library used, is the Libgcrypt that is developed as a separate module of GnuPG in C language. It provides functions for many fundamental cryptographic building blocks. Libgcrypt was also FIPS 140 validated which makes it a reliable source for running the desired tests. Algorithms that are tested using Wolfcrypt are: 3DES, RC4, Camellia, ChaCha, HC128, IDEA, and Rabbit, while the algorithms that are implemented using Libgcrypt are Arcfour (RC4), Blowfish, Camellia, ChaCha, and Twofish. The remaining codes (RC4D, LBlock, Present, Tea, XXTEA, Hight) were taken from well known programmers and trusted repositories.

5.3.1/ Proposed Scenario

In order to facilitate the implementations on both tools, the ”testingRNG” [371] project released by Daniel Lemire is used which aims at making it easier to run such tests on either MacOS or Linux with a recent C compiler. The seed used in the tests is generated by using the **”splitmix”** (a fast splittable PRNG) [323] which is widely used and is a part of the standard Java API. This generator produces 64 bit numbers. Then, we considered the **worst case scenario** where the whole plain text is just zeros and in each execution we tend to change only the key or the IV (depending on whether the algorithm has an IV or not) without changing anything else in the algorithm. The key has indeed a major effect on the randomness produced by the algorithm. Then, the tests will start running for a long time before the result can be seen in a log file in Practrand or by checking any failure in TestU01 after the launching of BigCrush test.

In fact, all the tested ciphers have undergone the SmallCrush test which is a very quick test, most commonly used to gauge if it is even worth running the heavier tests.

According to the tests executed, Practrand was the most efficient in revealing some weaknesses in the tested algorithms. The tests have been executed for at least 4 terabytes data on Practrand. As shown in Table 5.2, the algorithms that failed using Practrand are: **RC4 using both cryptographic libraries and ChaCha using Wolfcrypt library only.** Since RC4D proposed different versions of coding as mentioned before, more than one version was tested on both tools. The successful one was RC4D with both optimized and plain versions. While for **RC4Dkip, both optimized and plain versions failed the tests on Practrand and TestU01**.

	Algorithm	TestU01	Practrand	Error Type in Practrand
Libgcrypt	RC4	✓	✗	FPF
	ChaCha	✓	✓	-
	Camellia	✓	✓	-
	BlowFish	✓	✓	-
	TwoFish	✓	✓	-
WolfCrypt	RC4	✓	✗	FPF
	ChaCha	✓	✗	FPF GAP BCFN DC6
	HC-128	✓	✓	-
	Camellia	✓	✓	-
	IDEA	✓	✓	-
	Rabbit	✓	✓	-
	3DES	✓	✓	-
Git Repositories	Hight	✓	✓	-
	LBlock	✓	✓	-
	Present	✓	✓	-
	XXTEA	✓	✓	-
	TEA	✓	✓	-
	RC4D_plain	✓	✓	-
	RC4D_optimized	✓	✓	-
	RC4Dkip_plain	✗	✗	BCFN
	RC4Dkip_optimized	✗	✗	BCFN

Table 5.2: Successes and Failures obtained by Practrand and TestU01

5.3.2/ RESULTS INTERPRETATION AND DISCUSSION

The simplest way to interpret test results in Practrand is to look for the word ”FAIL” in the output. It will appear on the right-hand side and easy to be noticed.

The ciphers used produces a series of temporary result summaries as it goes along. Each result summary has a header showing the cipher tested, the number of bytes tested, the time taken, and the RNG seed used (so you can reproduce the results later if desired). The body of the result summary is a table showing all the irregular results followed by a statement of how many results were omitted from the table because they were regular. If the table would have zero entries then the table is skipped. If no results were omitted then the number of omitted results is skipped. The table of the result in

Practrand has four columns: (1) ”Test Name”, a name for the sub-test the corresponding, (2) ”Raw” not of much use to end users, (3) ”Processed”, either a p-value or ”pass” or ”fail”. (4) ”Evaluation”, describing the result. ”FAIL” means that the tested cipher un-vaguely failed that sub-test, while ”suspicious” means that that result should not happen often on a good RNG/cipher but should happen occasionally.

The failing results can be summarized as follows. RC4 in Libgcrypt failed Practrand after 1 terabytes of data indicating the error: FPF-14+6/16:cross. RC4 in Wolfcrypt also failed after 1 terabytes of data signaling the same error: FPF-14+6/16:cross. This clearly shows that the same issue in RC4 is found in both libraries. As it is known, RC4 does not have the required diffusion layer to increase the randomness of the output, and therefore, it has been breached. In fact, as mentioned before, FPF shows that there is a correlation in the ciphered output and that de-correlation fails. This can be clearly shown the Figure A.1.

For ChaCha the results in Wolfcrypt were disastrous in terms of randomness. All the tests that are available in Practrand failed after 256 gigabytes of data. GAP, FPF, DC6 and BCFN were all violated. In particular, a major failure that occurred in one of the oldest and most important tests was the GAP test [3]. The result is shown in Figure A.2.

For the new proposed diffused RC4, which is considered by the author as a chance of reviving the original breached RC4, the author succeeded in adding the new diffusion layer, and adding a substitution operation. The original versions of the code, the plain and the optimized ones, did well in Practrand and TestU01. However, the other versions that are based on adding some functionalities to the code, RC4Dkip, did not succeed neither TestU01 nor Practrand using both of their versions, the optimized and the plain codes. For Practrand, it failed after 4 terabytes of data showing the exact error in the plain and optimized versions which is the BCFN(2+0,13-0,T) error. Basically, BCFN tests for long range patterns in the distribution of 0s and 1s. So failing BCFN generally means that given whether 0s or 1s were more common in the previous 32 or 64 byte (n-1) blocks, you have enough information to guess which is more common in the nth with an accuracy that is above 50%, in this error n is 13. The result of Practrand is added in Figure A.3.

For TestU01, the value that enables us to decide whether this cipher succeeded the test or not is the list of p-value. If the values come outside the [0.01,...,0.99] interval, then this cipher fails TesuU01. When a p-value is extremely close to 0 or to 1 (for example, if it is less than 10^{-10}), one can obviously conclude that the generator fails the test. In this case a failure is is defined as a p-value $\leq$ (1.0e-10) or $\geq$ (1-1.0e-10). To see the summarized result of the failure in TestU01, there exists a file ./summarize.pl *.log that contains all the summarized crushed encountered during the test. The results can be seen in figure A.5 and the crushes are obvious. There were 10 crushes, in the ”msb” and ”lsb” tests.

After these tests, it can be deduced that whenever a cryptographer wants to propose a new algorithm, from a cryptographic point of view, it is important to test the new proposal using the simple methodology proposed. This will either highlight some flaws in the considered cipher or it can help in choosing/building a better cryptographic library that can be used without jeopardizing the security level.

5.4/ CONCLUSION

Randomness plays a major role in cryptography and the major goal for any cryptographer is to ensure the safety and the reliability of the proposed algorithm. Even if the majority of the implemented algorithms succeeded TestU01 and Practrand, it was shown that others did not. It is important to ensure the required randomness by availing all the tools that are available to do so. Validating the randomness of the ciphered outputs using the methodology proposed in this paper is quite simple yet very efficient. The approach proposed relies on using TestU01 and Practrand tests that are originally designed to test the randomness of RNGs. The approach was used to test the ciphered outputs in different well-known algorithms that are used in securing different multimedia content. The results obtained showed the following: RC4 in both WolfCrypt and Libgcrypt libraries as well as ChaCha implemented in WolfCrypt failed Practrand test. In addition, the new proposed RC4 with an additional diffusion layer failed TestU01 and Practrand in one of its coding versions, RC4Dkip. This interpretation of results provides the cryptographer with a way to validate the randomness of cryptographic ciphers before risking the security of data.

6

SPECK-R: AN ULTRA LIGHT-WEIGHT CRYPTOGRAPHIC SCHEME BASED ON SPECK

"Fully secure systems don't exist today and they won't exist in the future." — Adi Shamir

6.1/ INTRODUCTION

As discussed in the Scientific Background, securing data before sharing it across different platforms is a major necessity. However, new modern applications have a lot of limitations that need to be taken under consideration. Lower response time, high level of security and lower number of encryption rounds are all basic needs for any new proposed lightweight cipher.

SPECK, one of the ARX (Addition/Rotation/XOR) lightweight ciphers; proposed by the US National Security Agency (NSA) in 2013; offers security in constrained devices. It is well-known for its fast execution time, security and simple operations used. Speck and Simon were both proposed at the same time, however, Speck has been optimized for performance in software implementations, whereas Simon, has been optimized for hardware implementations. In this chapter, based on all the requirements that are previously explained, a new proposal based on the original Speck is explained. We propose an enhancement for the Speck algorithm, that needs a lower number of rounds and less execution time by adding a robust and dynamic level of substitution. The new proposed cipher ensures (a) confidentiality of the transmitted/stored data content in a robust way to protect it against attacks and (b) maintain a fast execution time in order to cope with the advanced demands of new devices.

This Chapter is divided as the following. The main features of the proposed image encryption algorithm are described in Section 6.2. Then in Section 6.3, a deeper look into Speck and its variants is presented. After that, Section 6.4 discusses the proposed cipher. Then, the added cryptographic layer is explained in Section 6.5 and specific tests are exerted to prove the robustness of the added substitution layer. After that, randomness tests done using Practrand are explained in Section 6.6. Security results that have been con-

ducted to evaluate the efficiency of this algorithm are also explained in Section 6.7. Then in Section 6.8, a performance analysis of Speck-R is presented and a comparison with the original Speck is given. A discussion about the efficiency of the proposed algorithm against the most known types of attacks is investigated in Section 6.9. Finally, Section 6.10 ends with a brief conclusion resuming the work.

6.2/ FEATURES OF THE PROPOSED APPROACH

Before digging deep into the original versions of Speck, the goals of the new updated Speck are described. We call our algorithm Speck-R where R stands for ”Reduced”. Speck-R meets two main contributions which are the high efficiency and the security compared to the original Speck. Below, the desired **system performance and security performance** are described.

System Performance:

- Lightweight: The minimum required number of iterations, for recent lightweight cryptographic algorithms, is 4 such as the Hummingbird2 cipher. For Speck, the minimum number of rounds is 22. In fact, Speck is based on ARX (Addition/Rotation/XOR) which is a class of cryptographic algorithms that has three simple arithmetic operations: namely modular addition, bitwise rotation and exclusive-OR. In both industry and academia, ARX cipher has gained a lot interest and attention in the last few years. By using combined linear (XOR, bit shift, bit rotation) and non-linear (modular addition) operations and iterating them for many rounds, ARX algorithms have become more resistant against differential and linear cryptanalysis. In this proposal, we aim at adding a dynamic substitution layer that increases the security of the cipher, yet keeps it ultra-weight. The proposed cipher avoids using a static diffusion operation such as the MixColumn transformation of AES [292] or the key-dependent integer/binary diffusion operations of [347, 376], since such operations consume a high percentage of the execution time [376, 324]. Moreover, Speck-R is realized in CTR mode, thus it can be processed in parallel, whether for encryption or decryption. CTR mode decreases the latency and enables a fast execution time.

- Flexibility: As the original Speck, it operates at the block-level, which can have a flexible number of bits exactly as the original cipher. This chosen block size can be set according to the user’s requirements and the network abilities. The proposed approach is set according to the devices’ characteristics.

- Simple hardware and software implementations: As stated earlier, ARX ciphers are easy to be implemented and are highly recommended for small, limited devices, especially those dedicated for IoT. This renders the corresponding hardware and software implementations to be simple and efficient.

- Low error propagation: In this proposal, each block is treated once at a time. The block is split into two parts, semi-blocks, thus, any error occurring in a block, will only affect the block itself. It will not affect the whole blocks in the image and the

error will not propagate across the whole image/data. Speck-R is designed to be in the CTR mode, thus avoiding any chaining process that itself propagates any error across the system. Thus, low error propagation is guaranteed.

- Large key space: Since the original Speck has different versions using different key sizes, the key can range between 64 and 256 bits. Therefore, adapting the same criteria of Speck, Speck-R is resilient against brute-force attacks according to [283].

These enhancements added to the cipher reduce the delay of the encryption and decryption processes and simplify their corresponding hardware implementations. Every primitive in this proposal has its own impact on the security and efficiency of the proposed cipher scheme.

Security Performance:

- Key dependent approach: Speck-R is based on key-dependent substitution primitive that ensures simplicity in addition to the required cryptographic properties.
- Dynamic key approach: Speck has already proven to be a secure cipher that possesses a secure key. We add to the cipher a dynamic substitution layer, that changes according to the number of iterations the cipher undergoes. The substitution layer is set to be dynamic, which means that it is built according to a previously chosen key. In contrast to the existing cipher solutions, the proposed approach is based on a dynamic key, which is variable and changes in a pseudo-random manner for each new session. The periodic interval of a session depends on the application or user requirements. For example, a new session can be established for each new input image. Therefore, the cryptanalysis process against the proposed cipher is very challenging because of the unpredictability of the cipher primitive as they change according to the dynamic key. Changing the key each time results in a different substitution layer that itself will change with the number of iterations. Adding a dynamic layer did not only result in having a more secure cipher, but it reduced the number of iterations from 26 to 7 which is the main goal behind this proposal.
- Speck original security: Until 2018, there were no published ”attacks” on full-round Speck but only on the reduced-round variety. These kinds of attacks aim to find the maximum number of rounds that will make Speck susceptible to theoretical attacks. As the designers of Speck claim, the cipher is designed to be resilient against standard chosen-plaintext and chosen-ciphertext attacks as well as related-key attacks. We can take the total number of rounds that have been attacked, as a percentage of the total number of rounds. As for 2018, there are no published works that attack more than 70-75% of the number of rounds through Speck. More than 70 papers are published, the best are 19 of 27 rounds for Speck 64/128. (70.3%) [309]. According to Speck’s original designers, they state that they made a trade-off between the security desired and the efficiency of the cipher, thus, we can say that Speck-R has the same properties as Speck. Based on the analysis done, stepping to appropriately balance efficiency and security has been reached. It can also be noted that the number of rounds considered in the Speck were based upon making it robust against differential attacks. They set the number of rounds to leave a security margin similar to AES-128’s at approximately 30% [355].

Accordingly, a good lightweight, flexible, cipher candidate based on Speck is proposed. This is justified since the trade-off between system performance and the security level is reduced in addition to its simple hardware and software implementations.

6.3/ DEEPER INTO SPECK

Speck and Simon were proposed by NIST in 2013 after seeing that traditional cryptography is no longer well-suited for the emerging reality. Speck and Simon are both block ciphers proposed to address the challenges in the constrained devices. They were first proposed by a group of researchers in 2013 [290] and many crypt-analysts worked since 2011 on proving that these new algorithms tend to be secure. Their results assured that Simon and Speck are both secure. The major issue is that most of the proposed lightweight ciphers lack the main criterion which is **flexibility**. This is typically what Speck and Simon aimed at. In fact, heterogeneous networks connect now a days millions of small devices, thus, the main aim is to ensure that the cipher will work properly and efficiently anywhere on any device. After all, we do not know what sort of new devices will exist in 2030. However, regardless what the device is capable of doing, it will sure support simple operations based on AND, OR, and XOR. These operations are done efficiently on small devices like FPGAs, since they aid at having a better performance for any proposed cipher. For example, PRESENT [141] which did great on ASIC did not perform well on constrained devices. Additionally, most of the proposed ciphers had fixed block size and key size. Additional flexibility is needed here, therefore Speck and Simon proposed using different sizes of blocks and keys. In this work, we chose to work on Speck, since it is more efficient than Simon in terms of software efficiency. Speck uses modular addition for its non-linearity, which is stronger in terms of cryptography than Simon's AND operation and is better suited in software implementation. In Speck, they generalized Feistel structure containing five different block sizes of 32, 48, 64, 96 and 128 which can be further divided into ten variants along with size key used. However, the flexibility and simple design ended up with algorithms that have exceptional performance on high-end platforms as well. Speck has the highest throughput on 64-bit processors of any block cipher implemented in software.

The round function consists of XOR, modulo addition and rotation operations. Below, a general round function of Speck is demonstrated in Figure 6.1 where L_i and R_i are the left and right half intermediate values respectively of the input for the i_{th} iteration. K_i is the n bit key used in the i_{th} round, α $<<<$ and β $<<<$ denote circular left and right shift by α or β bits, $\bigoplus$ is the XOR operation and $\boxplus$ is the modulo addition. The outputs for the i_{th} round are L_{i+1} and R_{i+1} and the round function can be described as follows:

$$L_{i+1} = ((L_i >>> \alpha) \boxplus R)) \oplus k_i \quad and \quad R_{i+1} = (R_i <<< \beta) \oplus L_{i+1}.$$

6.3.1/ THE DIFFERENT VERSIONS OF SPECK

In this subsection, the five different variants of the Speck family are represented. The rotation parameters (α, β) for Speck are either (7,2) or (8,3). In fact, Speck is usually denoted by Speck2n/mn where 2n is the block size and n $\in$ 16, 24, 32, 48, 64 and mn

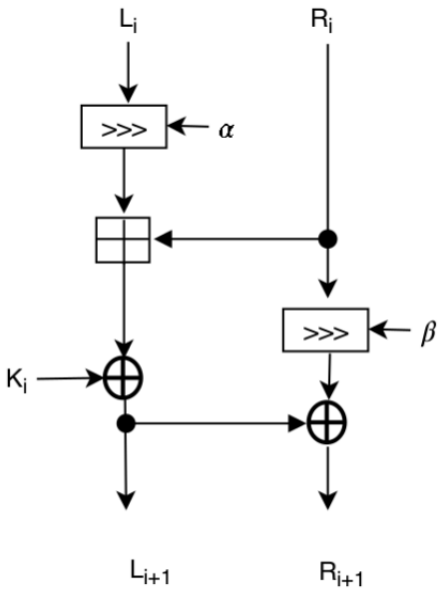


Figure 6.1: A representation for the general round of Speck cipher.

resembles the size of the key used where m $\in$ belongs to 2, 3, 4 depending on the desired security. Concerning the key schedule for the Speck family, the key used is 2, 3 or 4 words. The key schedule expands the initial m-word master key (l_{m-2},..., l_0, k_0) into i_{th} number of rounds (k_0, k_1,..., $k_{i_{th}}$), then two sequences are generated of words k_i and l_i according to the following algorithm:

$$l_{i+m-1} = ((k_i \boxplus (l_i >>> \alpha)) \oplus i \quad and \quad k_{i+1} = (k_i <<< \beta) \oplus l_{i+m-1}.$$

In Table 6.1, the different versions are represented. It can be clearly seen that the round function of Speck can have different key sizes and blocks and the number of rounds depends on the key used. In this work, we chose the version of Speck64/96 that operates in CTR mode with 26 rounds. In Figure 6.2 a detailed scheme of the round function with the key schedule is represented.

6.4/ THE PROPOSED SPECK-R

In this section, the proposed cipher algorithm is presented. First, we will begin by introducing and discussing some concepts used in our algorithm, then, the updates added to the original cipher are explained. Then, we provide details about the core of the ciphering layers used.

Block size (bits)	Key size (bits)	α	β	Number of Rounds
32	64	7	2	22
48	72	8	3	22
48	96	8	3	23
64	96	8	3	26
64	128	8	3	27
96	96	8	3	28
96	144	8	3	29
128	128	8	3	32
128	192	8	3	33
128	256	8	3	34

Table 6.1: Different parameters of Speck family.

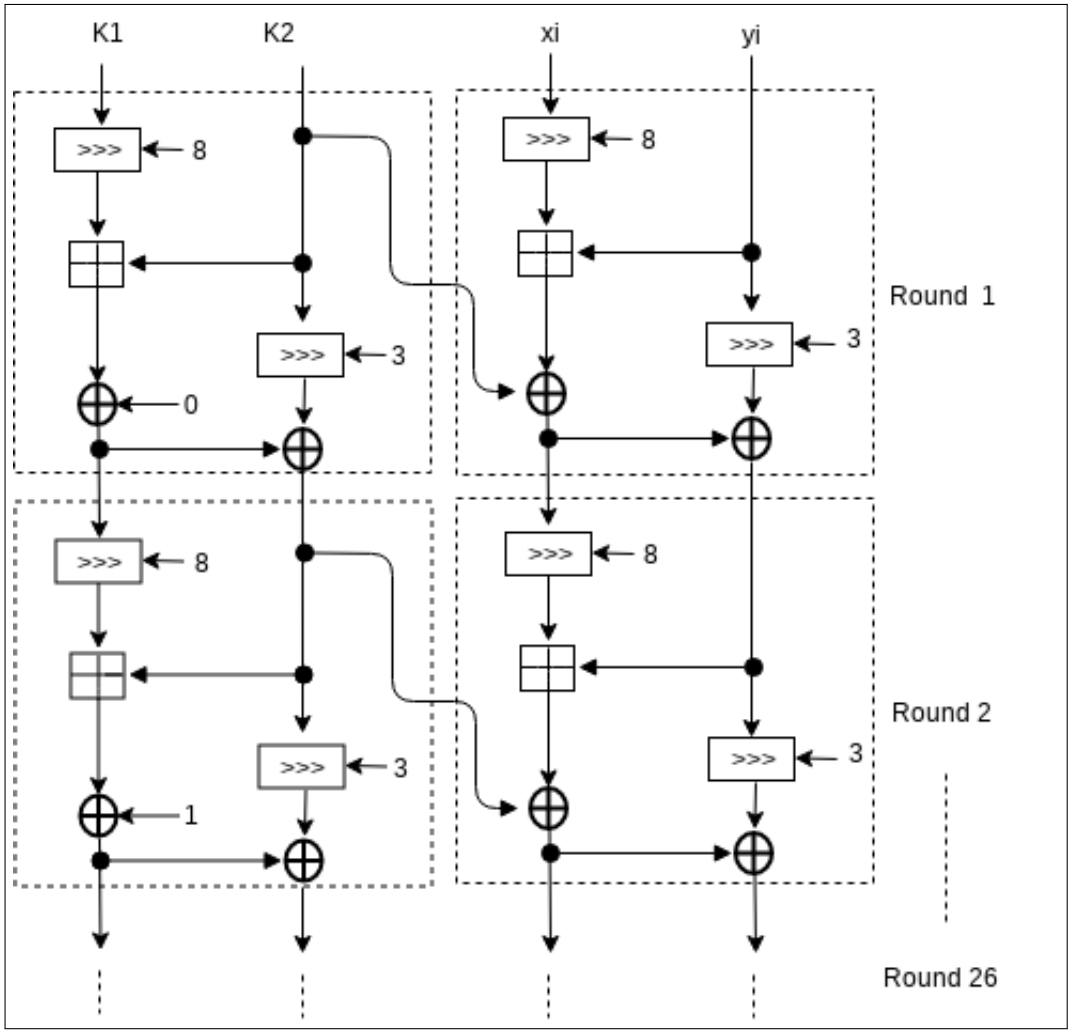


Figure 6.2: The original Speck64/96 cipher.

6.4.1/ KEY DERIVATION SPECK-R:

As can be seen, Speck lacks any substitution operation which they intended to do, to keep the algorithm as simple as possible. However, when iterating for 22 rounds or more

(in our case 26 rounds), in our opinion, this can be reduced to minimize the execution time. What we aim at, is to optimize as much as possible the number of rounds used in the cipher.
The proposition falls within the symmetric key schemes where the two communicating parties (sender, receiver) share the same key. The main advantage behind this is the low complexity compared to the public key schemes. Parameters that are needed in Speck-R are: a Nonce N, a Dynamic key DK and the Key K. All the notations used are shown in Table 6.2 and the initialization phase is demonstrated in Figure 6.3. These steps are sufficient to preserve high sensitivity since a little change will lead to completely different parameters and substitution tables.

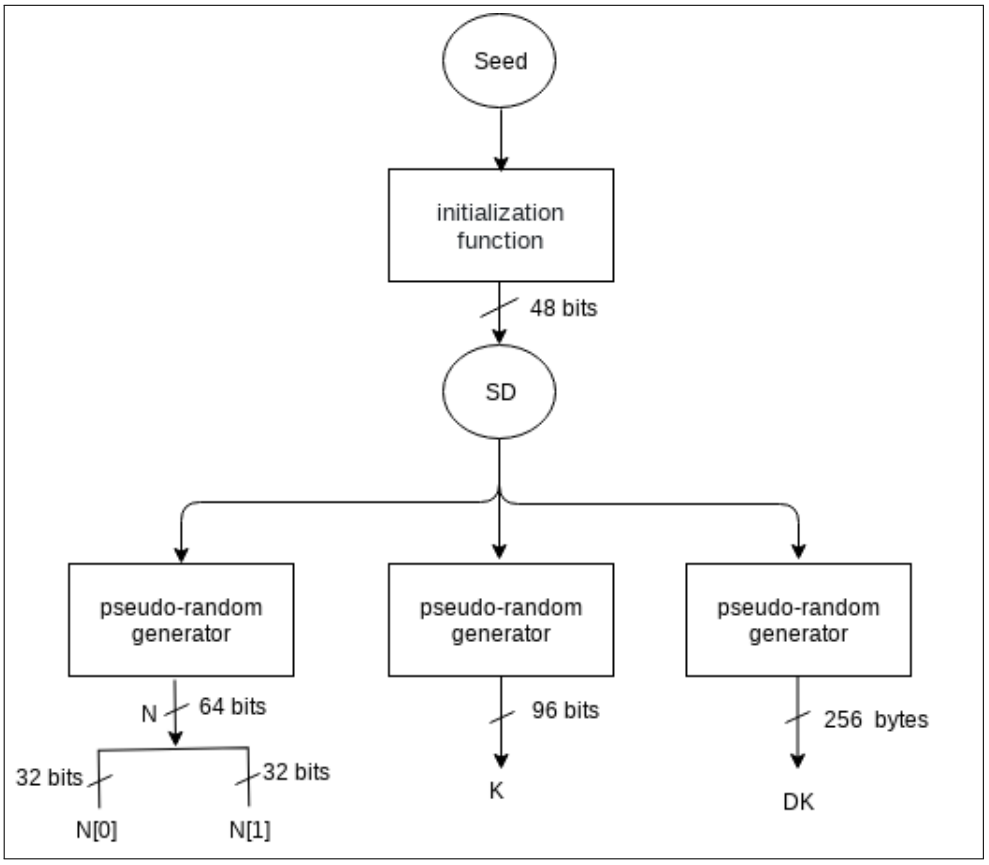


Figure 6.3: Keys and parameters required in the proposed Speck-R.

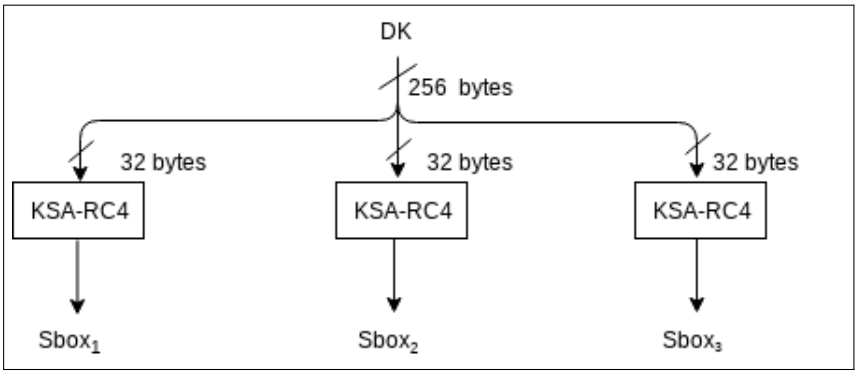


Figure 6.4: A high level scheme for the Sboxes generation.

- Initialization Function: It is a function used to generate the seed that will be used later on in the pseudo-random generators. It can be any function the user defines, a hash function (like SHA-512), or a key stream cipher. A hash function will be useful here, to avoid any collisions, but it will be expensive in terms of time. In our scope, we intended to use a simple pseudo random generator yet known to be efficient, which is Splitmix64. It is a split-table pseudo-random number generator that is based on object-oriented arithmetical and logical operators [323].

- Nonce: Denoted by N, which is needed in any counter mode cipher. A pseudo-random generator is used to generate this Nonce from a seed. It is important to generate a new Nonce for each input image. N can be sent to the receiver encrypted using the shared public key of the other entity if the asymmetric approach is used. Another way for sharing N is to have a good synchronization between the sender and the receiver where each entity derives it separately with no need for transmission and starting from the same seed. In Speck-R, Nonce has a length of 64 bits (8 bytes). The 64 bits are split into parts with 32 bits, where the first 32 bits represent N[0] and the second 32 bits represent N[1]. After the generation of the Nonce, N[1] will remain the same, whereas N[0] will be incremented by 1 after every iteration (i.e. from one block to another).

- Key: Denoted as K, which will be used as the main key to extract the round function keys, just as Speck. A pseudo-random generator is used to generate K. The same key schedule as in Speck is used, that is, the key generated which is 96 bits will be expanded. In every round of Speck-R a unique key will be used which is derived from K. However, in the original version of Speck, there were 26 round different keys, while in the proposed version there are only 7 round keys, each of 32 bits, denoted by K_r.

- Dynamic Key: Denoted as DK is also generated by using a pseudo-random generator. It has a length of 256 bytes, which will be later on used to generate three different substitution boxes that are used in the encryption process.

- Substitution Boxes: Denoted as $Sbox_1$, $Sbox_2$, $Sbox_3$. The cryptographic strength of three Sboxes will be explained in Section 6.5 more clearly. However, to generate these three Sboxes RC4 will be used. RC4 is not used here in the context of a stream cipher, that mixes the plain text with the output key stream. It is iterated according to the DK previously produced to generate three robust Sboxes. RC4 is used since it is well known for its simple hardware and software implementation. Key Setup Algorithm (KSA) which is the initialization phase of RC4 is used in specific to generate the three dynamic Sboxes. This is demonstrated in Figure 6.4, and the algorithm is shown in Algorithm6.1.

6.4.2/ Encryption Process

In general, Speck can operate in different encryption modes i.e. ECB, CTR, CBC, PCBC, CFB and OFB. The proposed Speck-R operates in CTR mode, however, it can be executed in other modes. Speck-R can be used for the encryption of any kind of data whether texts or images etc.. In the case of image encryption, the image is of size $M \times N \times P$ where M is the columns number, N is the rows number and P is the plane number (for grey-scale

Algorithm 6.1 KSA for RC4

procedure RC4_KSA($K = \{k_1, k_2, \ldots, k_L\}, L$)
  for $i \leftarrow 0$ to 255 **do**
    $S\ [i] \leftarrow i$
  end for
  $j \leftarrow 0$
  for $i \leftarrow 0$ to 255 **do**
    $j \leftarrow (j + S[i] + k[j \bmod L]) \bmod 256$
    $swap(S[i], S[j])$
  end for
  return S
end procedure

Table 6.2: Summary of the notations used.

Notation	Definition
$Seed$	Seed used as an input for an initialization function
SD	Seed used as an input for the pseudo-random generators
N	Nonce 64 bits
$N[0]$ *or* N_R	First 32 bits of N
$N[1]$ *or* N_L	Second 32 bits of N
K	Key of 96 bits used in Key schedule of Speck-R
K_r	The key used in every round
DK	Dynamic key of 256 bytes used to build $Sbox_1$, $Sbox_2$, $Sbox_3$
$Sbox_1$	The first produced dynamic substitution table
$Sbox_2$	The second produced dynamic substitution table
$Sbox_3$	The third produced dynamic substitution table
Seq_i	The plain block at index i
X_L	The encrypted N_L
Y_R	The encrypted N_R
n	Number of bytes in the block
nb	Number of blocks present in the plain message
M	Number of columns of an image
N	Number of rows of an image
P	Number of plane (in gray-scale P=1)

is equal to 1). Image is stored using Pixmap that stores and displays a graphical image as a rectangular array of pixel color values [62]. The general encryption process in the CTR mode is displayed in Figure 6.5. As can be seen, Speck-R takes one block Nonce of 64 bits, and this Nonce is split into two blocks of 32 bits. The result of the encrypted 32 bits N[0] and N[1] after passing through Speck-R will be xored with 32 bits block of the input. The encryption continues to cover all the blocks in the original plain-text/image. Let n represent the number of bytes chosen in the block, according to each version of Speck, then the total number of blocks nb will be equal to

$$nb = \lceil \frac{(M \times N \times P)}{n} \rceil \tag{6.1}$$

where the index of blocks i, will be lie between $i \in \{0, 1, ..., nb\}$.

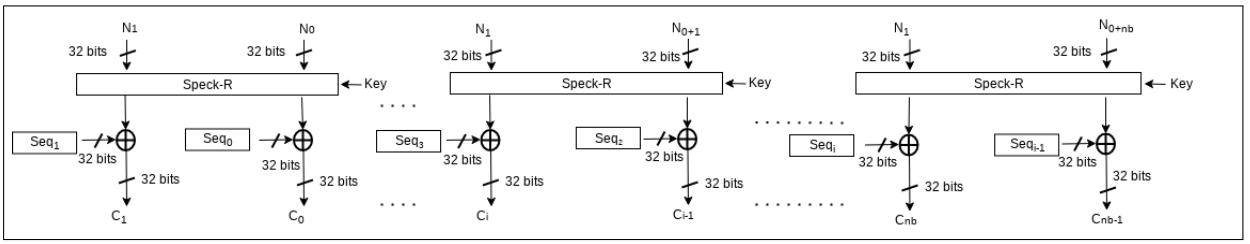


Figure 6.5: The general scheme of the proposed cipher Speck-R.

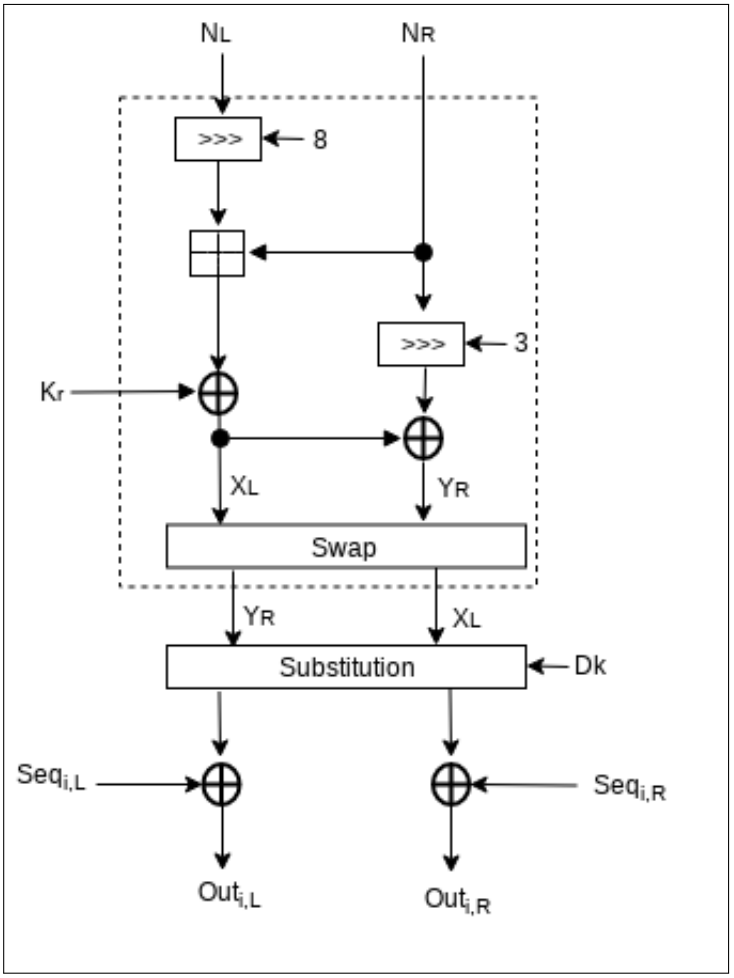


Figure 6.6: The proposed encryption round of the cipher Speck-R.

In the rest of this work, the size of the chosen block is 64 bits, and the key is chosen to be 96 bits. In Figure 6.6 a closer view on the round of Speck-R is represented. The proposed encryption algorithm can be divided into three major operations which are, (1) Encrypting the Nonce, (2) Passing across the substitution layer, and (3) Xoring the plain text with the resulted substituted value. The following operations are explained as follows.

1. Encrypting the Nonce: After the generation of N, it will be divided into two smaller blocks, $N[0]$ and $N[1]$ that are denoted by N_R and N_L. Each block will undergo the same round function of Speck that is represented by the following equations:

$$X_L = ((N_L >>> 8) \boxplus N_R)) \oplus K_r \tag{6.2}$$

$$Y_R = (N_R <<< 3) \oplus X_L. \tag{6.3}$$

As can be seen, this step is just an ordinary Speck round, where K_r, is the key to be used for every round and $K_r \in \{ K_1, K_2, K_3, K_4, K_5, K_6, K_7 \}$. This key is generated by using the key expansion method used in Speck.
After the encryption of N_L and N_R, Speck-R swaps the two outputs X_L and Y_R. That is to say that X_L will take the place of Y_R and vice versa. This will increase the randomness in the encrypted nonce and will lower the probability of any sequential relation with the next block, if it exists. $N[1]$ will remain constant throughout all the blocks, however, $N[0]$ will be incremented by 1 from one block to another. That is to change the Nonce value from one block to another, which adds more immunity to the cipher. Many ciphers in CTR mode, use a static Nonce, and this is not the case in Speck-R.

2. Substitution Layer:
The main component of Speck-R is the added substitution dynamic layer. As explained earlier, the DK will be used to generate three different substitution boxes ($Sbox_1$, $Sbox_2$, $Sbox_3$), by using the Key scheduling Algorithm (KSA) for RC4. In fact, a substitution table is a non-linear component added to the cipher to attain the confusion property, as explained earlier in this thesis. We propose to use a dynamic substitution layer that is built upon a dynamic key that changes for every input. At the beginning of the encryption, two counters are initialized, $it_1 = 0$ and $it_2 = 0$. The counters will be incremented in every round by one element. When it_1 reaches 2000, then the substitution table $Sbox_1$ will be replaced by a the resultant of the substitution operation of $Sbox_1$ by $Sbox_2$. In other words, by using $Sbox_2$, $Sbox_1$ will undergo a substitution operation. This can be represented as $Sbox_1 = Sbox_2[Sbox_1]$. Then, it_1 is set back to 0. If the number of blocks was very large, and it_2 reaches a value of 2000×2000, then, $Sbox_2$, will be subjected to a substitution operation, using $Sbox_3$. This will be represented as $Sbox_2 = Sbox_3[Sbox_2]$. The following steps are explained in Algorithm 6.2. The main aim of adding this layer is decreasing the round number of Speck from 26 to 7 in Speck-R.

3. Xor the plain-text: As in any CTR cipher, the ciphered final result, is the plain text xored with the encrypted Nonces/counters. After passing through the substitution layer, the plain text/image will be divided into blocks each of 64 bits. Then, the first and second 32 bits, $Seq_{i,L}$ and $Seq_{i,R}$, will be xored with the substituted values of Y_R and X_L, respectively. This is represented by the following equation:

$$Out_{i,L} = Seq_{i,L} \oplus Sbox_1\ [Y_R]; \tag{6.4}$$

$$Out_{i,R} = Seq_{i,R} \oplus Sbox_1\ [X_L]; \tag{6.5}$$

Finally, to get the whole result, all the blocks are concatenated, and aligned using the Pixmap. The encryption is simple, efficient, dynamic and easy to be implemented. As a conclusion, this is a simple cipher that reaches the confusion and diffusion properties with just 7 rounds via a dynamic key dependent substitution

Algorithm 6.2 Dynamic Substitution Layer

procedure SUBSTITUTION($\{Sbox_1, Sbox_2, Sbox_3\}$)
  for $i \leftarrow 0$ to nb **do**
    $Encrypt(Block\ [i])$
    $it_1 \leftarrow it_1 + 1$
    $it_2 \leftarrow it_2 + 1$
    if it_1 =2000 **then**
      $Sbox_1 \leftarrow Sbox_2\ [Sbox_1]$
      $it_1 \leftarrow 0$
      if it_2 =2000×2000 **then**
        $Sbox_2 \leftarrow Sbox_3\ [Sbox_2]$
        $it_2 \leftarrow 0$
      end if
    end if
  end for
end procedure

layer. The efficiency and robustness are demonstrated in the following sections. The whole encryption algorithm can be summarized in Algorithm 6.3.

Algorithm 6.3 Encryption Process of Speck-R

procedure SPECK-R_ENCRYPTION(Seq)
  $N \leftarrow PRNG\ (seed)$
  $K \leftarrow PRNG\ (seed)$
  $K_r \leftarrow Key\ Expansion$
  $DK \leftarrow PRNG\ (seed)$
  $Seq\ [nb] \leftarrow plaintext$
  $Sbox_1,\ Sbox_2, Sbox_3 \leftarrow RC4 - KSA\ Initialization\ Algorithm$
  for $i \leftarrow 0$ to nb **do**
    $X_L = ((N_L >>> 8) \boxplus N_R)) \oplus K_r$
    $Y_R = (N_R <<< 3) \oplus X_L$
    $Swap(X_L, Y_R)$
    $N_L \leftarrow N_L$
    $N_R \leftarrow N_R\ ++$
    $Out_{i,L} \leftarrow Sbox_1\ [Y_R] \oplus Seq_{[i],L}$
    $Out_{i,R} \leftarrow Sbox_1\ [X_L] \oplus Seq_{[i],R}$
    $Update\ Sbox_1 \leftarrow Substitution\ (Sbox_1,\ Sbox_2,\ Sbox_3)\ Algorithm$
  end for
end procedure

6.4.3/ DECRYPTION PROCESS

Similarly for the encryption process, the decryption process needs 7 rounds. The ciphered scheme will be xored with encrypted Nonces. There is no need to use an inverse substitution layer, since the same substitution tables will be used. The decryption process is exactly the same as the encryption which gives a great advantage to the scheme

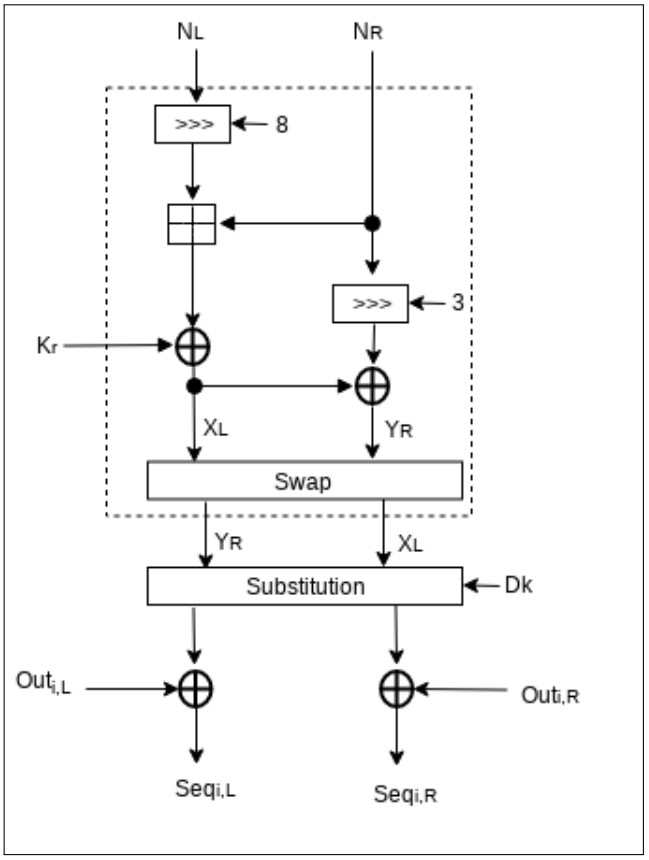


Figure 6.7: The proposed decryption round of the cipher Speck-R.

in terms of software/hardware implementation. No complicated re-evaluation for the inverse substitution tables is needed here. The decryption process is shown in Figure 6.7. In the next sections, the robustness of Speck-R will be proved and extensive tests are executed.

6.5/ CRYPTOGRAPHIC STRENGTH OF CIPHER LAYERS

In this section, we evaluate the performance of the proposed dynamic layer. In general, the substitution operation is used to ensure the confusion property and to introduce non-linearity in any cipher scheme. The proposed cipher needs three substitution tables: $Sbox_1$, $Sbox_2$, and $Sbox_3$. Mainly, $Sbox_1$ is used directly onto data encrypted/decrypted, while the other two $Sboxes$ are used to manipulate the first Sbox. $Sbox_2$ is used to change $Sbox_1$ after a predefined number of iterations and $Sbox_3$ is used to alter $Sbox_2$ after another predefined number of iterations. As mentioned before, the initialization phase of RC4-KSA, is used to generate the dynamic substitution layer [29]. It is described in Algorithm 6.1, where the dynamic input key DK with length L bytes is introduced to produce the three substitution tables. In the work presented, the first 32 bytes are taken as input to produce $Sbox_1$, the second 32 bytes produce $Sbox_2$ and the third 32 bytes of DK results in $Sbox_3$. The size of the produced substitution tables is 256 element, which is 32 bytes. However, to demonstrate a strong substitution layer, based on information theory analysis [18, 26, 32], four main properties have to be insured which are (a) Linear Prob-

ability approximation Boolean Function (LPF), (b) Differential Probability approximation Function (DPF), (c) Strict Avalanche Criterion (SAC) and (d) output Bits Independence Criterion (BIC).

6.5.1/ Linear Probability Approximation Boolean Function (LPF)

LPF was first introduced in [32], in the proposition of a linear cryptanalysis for DES block cipher. The basic idea behind it, is to find a linear relation or approximation that relates some bits of the plain-text $\{p1, p2, p3, ..., pb\}$ with its corresponding ciphered ones where b represents the number of bits. Finding a linear relation between the plaintext and the ciphertext will make the key exposed and easier to be extracted $\{k1, k2, k3, ..., kb\}$.

Definition 3: LPF Form 1

For a substitution layer $F : [0, 2^n - 1] \rightarrow [0, 2^n - 1]$, the linear probability Boolean function is defined as the following:

$$LPF = Max_{\alpha,\beta\neq 0}\lceil(LPF_{(\alpha,\beta)})\rceil = Max_{\alpha,\beta\neq 0}[\frac{card\{i/i\odot\alpha = F(i)\odot\beta\} - 2^{n-1}}{2^{n-1}}]^2 \quad (6.6)$$

where $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ and $\beta = \{\beta_1, \beta_2, ..., \beta_n\}$, $\alpha, \beta \in [1, 2, .., n - 1]$, $card$ represents the cardinal and $F(i) \odot \beta$ represents $F(i)_1 \bigwedge \beta_1 \oplus F(i)_2 \bigwedge \beta_2 \oplus .. \oplus F(i)_n \bigwedge \beta_n$ and finally, $i \odot \alpha = i_1 \bigwedge \alpha_1 \oplus i_2 \bigwedge \alpha_2 \oplus ... \oplus i_n \bigwedge \alpha_n$.

Another form of equation 6.6 to represent LPF is as the following:

Definition 4: LPF Form 2

$$LPF_{(\alpha,\beta)} \neq \frac{1}{2^n - 1} \quad (6.7)$$

Otherwise, $\sum_{\alpha=1}^{2^n-1} LPF_{(\alpha,\beta)} = 1 \; \forall \; \beta$ and $\sum_{\beta=1}^{2^n-1} LPF_{(\alpha,\beta)} = 1 \; \forall \; \alpha$. All this means that the substitution layer's immunity is directly related to the uniformity of the LPF(α, β). The lower the value of LPF, the higher the complexity of linear attacks and vice versa. As an example, AES cipher has an LPF of $2^{-6} = 0.015625$.
In the proposed cipher, LPF was tested to prove that a low probability exists [32]. To reach a better resistance against linear attacks, LPF should be very low. In order to evaluate the required number of necessary iterations to attain the lowest LPF value, LPF values versus the number of iterations were tested. For each iteration, the computed number corresponds to the mean of 1000 tested sub-matrices. Results showed that after 4 iteration, LPF stabilizes and reaches its minimum which is $2^{-4.8} = 0.035897$. Consequently, it can be said that the substitution layer becomes immune against linear attacks after 4 iterations.

6.5.2/ Differential Probability Approximation Function (DPF)

Differential Probability is one of the important properties of any substitution layer for obtaining the nonlinear transformation, and hence resisting differential cryptanalysis attacks [26]. In fact, this criterion studies the effect of a slight change in plaintext pairs on the corresponding ciphertext pairs. The cryptanalyst in this attack tries to leverage the high probability of occurrence that appears in the difference of two plaintexts. The substitution layer must have differential uniformity. Particularly, the difference between two plaintexts $\triangle_{ik}$ must produce a unique difference in the output ciphertexts $\triangle_{fk} = F(i) \oplus F(i + \triangle_{ik})$.

Definition 5: DPF

DPF is defined as the following:

$$DPF = Max_{\triangle_i \neq 0, \triangle_F}[DPF(\triangle_i, \triangle_F)] \tag{6.8}$$

where

$$DPF(\triangle_i, \triangle_F) = \frac{card\{i/F(i) \oplus F(i + \triangle_i) = \triangle_f\}}{2^n} \tag{6.9}$$

where $\triangle_i \in [1, 2^n - 1]$ and $\triangle_f \in [0, 2^n - 1]$. In this work, DPF was calculated versus different number of iterations. For each iteration, the computed number corresponds to the mean of 1000 tested sub-matrices. Results showed that to provide a better resistance against differential attacks, minimum 4 iterations are needed so that the average of DPF converges to the minimum possible value $2^{-4.5} = 0.044194$.

6.5.3/ Strict Avalanche Criterion (SAC)

Webster and Tavares were the first to present SAC when they generalized the avalanche effect [18]. Referring to Shannon, an efficient cipher must ensure confusion and diffusion properties. That is to say, a cipher system function is satisfying SAC whenever a single input bit is complemented, the output bit should be changed at least with a probability of half.

Mathematical steps:
In order to calculate the SAC property, the following has been made. First, assume that the plaintext with n bits, is substituted using the non-linear function $F(i)$ where $i \in [0, 2^n - 1]$. For each input, these six steps are done:

1. The plaintext is arranged to be in the form in one vector denoted as $i = \{i_1, i_2, .., i_n\}$ and $i_k = \{i_1, i_2, \bar{i_k}, .., i_n\}$ where i and i_k are the same, except for the toggled bit at the $k - th$ index (i_k).

2. The non-linear function F is applied on i and $i_k$to produce $F(i) = \{F(1), F(2), .., F(n)\}$ and $F(ik) = \{F(1), F(2), ..F(\bar{k}), F(n)\}$.

3. Then a new vector V is defined as $V = [V(1), V(2), .., V(k), V(n)]$ where $V(k) = F(i) \oplus F(ik)$.

4. Then, $a_{j,k} = a_{j,k} + v_{j,k}$, where $j, k \in [1, 2, .., n]$. $v_{j,k}$ represents the j^{th} bit of vector v in its binary form and $a_{j,k}$ is the j^{th} element of the matrix of dependence A (defined as all zero elements) of size $n \times n$. $a_{j,k}$ demonstrates the relation between the bit k of the plaintext and its corresponding substituted bit j.

5. Then, the SAC matrix is calculated by dividing each element of matrix A by 2^n.

6. Finally, to say that the substitution matrix attains the SAC criterion, the mean of the matrix A must be close to 0.5.

In this work, we targeted 1000 sub-matrices to check if the substitution level attains the SAC criterion or not. The result obtained is that after 4 iterations, the produced $Sboxes$ become more close to the ideal value. Thus, we can say that after 4 iterations the cipher will be sensitive to any bit toggling and therefore, the avalanche effect is ensured in the level of substitution.

6.5.4/ Output Bit Independence Criterion (BIC)

This criterion measures the level of dependence of the output bits, after they undergo the substitution process defined by [18, 22]. According to this criterion, the inversion of an input bit p modifies output bits q and r without any dependence on each other. An S-box that makes the output bits independent of each other strengthens the security. It is calculated as the following:

Mathematical steps:
In order to calculate the BIC property, the following has been made. First, assume that the plaintext with n bits, is substituted using the non-linear function $F(i)$ where $i \in [0, 2^n - 1]$. For each input, these six steps are done:

1. The plaintext is arranged to be in the form in one vector denoted as $i = \{i_1, i_2, .., i_n\}$ and $i_k = \{i_1, i_2, \bar{i_k}, .., i_n\}$ where i and i_k are the same, except for the toggled bit at the $k - th$ index (i_k).

2. The non-linear function F is applied on i and $i_k$to produce $F(i) = \{F(1), F(2), .., F(n)\}$ and $F(ik) = \{F(1), F(2), ..F(\bar{k}), F(n)\}$.

3. Then a new vector V is defined as $V = [V(1,1), V(2,2), .., V(j,k), V(n,n)]$ where $V(k) = F(i) \oplus F(ik)$; $j, k = \{1, .., n\}$ and $j \neq k$.

4. Then, $b_{j,k} = b_{j,k} + d_{j,k}$, where $d_{j,k}$ is the Hamming distance of $V(j,k)$ in bits, and $b_{j,k}$ is one element in the matrix of dependence B (initially defined with zeros elements) of size $n \times n$ and it represents the relation between the substituted bit j and the substituted bit k.

5. Then, the SAC matrix is calculated by dividing each element of matrix B by 2^n.

6. Finally, to say that the substitution matrix attains the BIC criterion, the mean of the matrix B must be close to 0.5.

To prove that Speck-R meets the BIC criterion, 1000 different sub-matrix were used and after four iterations, the BIC becomes very close to the desired value 0.5. This

literally means that the two output bits j and k for each substituted bytes, will change independently if a single bit i is changed. Hence, under this value, the proposed substitution layer becomes immune against chosen plaintext/ciphertext attacks.

All the evaluated criteria show that four iterations are needed to reach the desired cryptographic strength. All the previous calculated values are presented in Table 6.3, where a comparison with the AES substitution table is made. The results showed that the proposed substitution layer possesses sufficient cryptographic performances and the obtained results of LPF, DPF, SAC and BIC are very close to the standardized solutions. The cryptographic security of our scheme relies on the property of using a new dynamic efficient substitution layer.

Test	Speck-R	AES
LPF	$2^{-4.8}$	2^{-6}
DPF	$2^{-4.5}$	2^{-6}
SAC	0.5	0.4998
BIC	0.51	0.4998

Table 6.3: A comparison analysis of the substitution layer of Speck-R and AES.

6.5.5/ VALIDATION BY THE SBOX EVALUATION TOOL

Parameters tested	AES [292]	PRESENT [141]	Klein [242]	RC4-KSA
Input size M	8	4	8	8
Output size N	8	4	8	8
S-box Balanced	Balanced	Balanced	balanced	Balanced
Correlations immunity	0	3	0	0
Algebraic immunity	4	0	0	4
Transparency order	7.860	4	0.486	7.797
Propagation characteristic	0	0	0	0
Robustness to differential cryptanalysis	0.984	0.984	0.004	0.953
SNR (DPA) (F)	9.600	0.250	0.133	8.73

Table 6.4: A comparison between SET execution samples: AES, PRESENT, KLEIN, RC4-KSA.

Robustness evaluation of $Sbox$ is not limited to those four criterion. In fact, there exist different tools to evaluate the robustness of the cryptographic substitution tables. The evaluation of substitution tables has been quite a difficult issue for researchers, since, the public available tools are few. Some of these tools are: (1) **Boolfun package in R** that works under Unix and the package named boolfun can be loaded for functionality related to cryptography [273, 305, 382]. (2) **Boolean functions** in Sage [186] is another tool to evaluate the S-box. It is a free and open source mathematics software, that is mainly used to evaluate cryptographic properties of Boolean functions, mainly related to linear and differential properties. (3) The third tool is actually a **module for S-box in Sage**, which only has the possibility of calculating the difference distribution table and the linear approximation matrix, in terms of cryptographic properties. (4) **VBF (Vector Boolean Functions) library**, which is not available on-line to use freely, is

presented by Alverez-Cubero and Zufiria for analyzing vectorial Boolean functions from cryptographic perspective that possibly could calculate various properties of S-boxes [212]. Finally, (5) **SET** (Sbox Evaluation Tool), which is available freely online http://sidesproject.wordpress.com/, proposed in 2016 and takes almost all the necessary criterion to evaluate the substitution table.
In this work, we first intended to use our own generated code to test the previously explained four criterion (LPF, DPF, SAC and BIC), and then validate the robustness of the proposed dynamic Sbox using the newest tool available, SET-tool. In Table 6.4, we give few examples for Sboxes of AES, PRESENT, KLEIN, and the proposed dynamic RC4-KSA. M and N represent the input and output variables of Sboxes. A subset of tests of the available properties is shown, which were executed on a machine with Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 16 GB RAM, and Linux Debian 9 (stretch).
The chosen subset of tests are listed below, where any curious reader can have the following references to dig deeper into these tests:

Balancedness: [118, 124] A Boolean function is balanced if its output is equally distributed, its weight is equal to 2^{n-1}. This can be translated as $W_f(0) = 0$ for the Walsh spectrum. In the test executed, the four Sboxes are balanced and satisfy this property.

Correlation Immunity: [115, 124] A function f is said to be correlation immune of order t, denoted by CI(t), if the output of the function is statistically independent of the combination of any t of its inputs. For the Walsh spectrum, it holds that $W_f(\bar{w})$=0, for $1 \leq wt(\bar{w}) \leq t$. In the tests executed, the correlation immunity was 0 in AES, KLEIN and RC4-KSA, whereas for PRESENT, it was 3 which is not the desired value.

Algebraic immunity: [124, 239] High nonlinearity is a necessary condition to resist algebraic attack and the value of algebraic immunity should not be low according to the study made in [116]. The core of the analysis is to find out minimum (or low) degree annihilators of f and $1 + f$, i.e., to find out minimum (or low) degree functions g_1, g_2 such that $f * g_1 = 0$ and $(1 + f) * g_2 = 0$. To start the algebraic attack, one needs only the low degree linearly independent annihilators of $f, 1 + f$. Boolean functions used in crypto-systems must have high non-linearity to prevent linear attacks [27]. In the results obtained, AES and RC4-KSA have the same algabraic immunity, 4, where as for KLEIN and PRESENT, it is 0.

Transparency Order (TO) : [169] Transparency order (TO) is the only one currently available to evaluate the inability of an S-Box to thwart the DPA attack. It has been proved that the smaller the TO of an S-Box, the higher its resistance would be against the DPA attacks. The value we obtained for RC4-KSA (7.797) is close to the TO of AES (7.860), however, it seems that KLEIN and PRESENT have a better TO, 0.486 and 4, respectively.

Propagation characteristic: [124, 25]: In general, a function is said to satisfy the propagation characteristics of degree l, denoted by $PC(l)$, if all its derivatives w.r.t. vectors $\bar{w}$ with $1 \leq wt(\bar{w}) \leq l$ are balanced. In the auto-correlation spectrum, this means that $r_f(\bar{w}) = 0$ for all $1 \leq wt(\bar{w}) \leq l$. The Sbox F is said to satisfy the propagation characteristic with respect to $\bar{a} \in \mathbb{F}_2^n$ (the set of all n-tuples of elements in the field $\mathbb{F}_2$,

Galois field with two elements) if and only if $F(\bar{x}) \oplus F(\bar{x} \oplus \bar{a})$ is balanced. The value of interest related to the differential table is called the differential uniformity, denoted by $\triangle(F)$: $\triangle(F) = \max_{a \neq 0, b} D_{a,b} = \max_{\bar{a} \neq \bar{0}, \bar{b}} \#\{\bar{x} \in \mathbb{F}_2^n : F(\bar{x} \oplus \bar{a}) \oplus F(\bar{x}) = \bar{b}\}$.
As it can be seen in Table 6.4, AES, PRESENT, KLEIN and RC4-KSA has 0 propagation. In fact, diffusion is related to the propagation characteristic. A major link between diffusion and confusion criteria was pointed out by Meier and Staffelbach [21]. They proved that maximal non-linearity and perfect propagation characteristics are equivalent requirements for Boolean functions with an even number of variables. Unfortunately those functions which achieve perfect diffusion and perfect confusion (called bent functions) are not balanced; that means that they do not have a uniform output distribution. The construction of balanced Boolean functions having a high nonlinearity and good propagation characteristics then remains an open problem although such functions are essential components of cryptographic primitives.

Robustness to differential cryptanalysis: [33, 26] It is at least $1 - 2^{-t}$, robust against differential cryptanalysis, where t is a parameter satisfying the condition that $[(s - \lfloor n/2 \rfloor) \geq t \geq 3]$. An Sbox attains its maximum robustness when $1 - 2^{-t}$ is minimum. In fact, to say that the following Sbox $(n \times s)$ is robust against differential analysis, the result obtained must be close to the above boundaries of the following $(1 - 1/2^n)(1 - 2^{-s+1})$. For AES 8×8: the upper boundary is: $(1 - 1/2^8)(1 - 2^{-8+1}) = 0.988$, which is very close to value obtained 0.984, then, AES is robust against differential analysis. For PRESENT, it is the same. While for KLEIN, the upper boundary is 0, the value obtained is close to 0. For RC4-KSA, the $Sbox$ used is 8×8, thus, the value obtained (0.953) is very close to the desired value (0.988).

DPA (Differential Power Analysis) Signal-to-noise ration SNR: In [103], authors showed that the DPA signal-to-noise ratio increases when the resistance of the substitution box against linear cryptanalysis increases. Mainly, secret key algorithms consist in the repetition of several rounds, and are thus threatened by the differential power analysis (DPA). The obtained SNR are 9.6 and 8.73 which are considered good to face the differential attacks, whereas the SNR for PRESENT and KLEIN are very low, 0.25 and 0.133 respectively.

The obtained results were sufficient to indicate that the proposed construction technique of key-dependent substitution produces a robust and efficient substitution table (Sbox). Furthermore, $Sbox_1$ $Sbox_2$, and $Sbox_3$ make the proposed cipher algorithm immune against differential and linear attacks, since they are changed in a pseudo-random manner.

6.6/ RANDOMNESS TEST VALIDATION

As indicated in the Chapter 3, we propose leveraging from TestU01 and Practrand to validate the level of randomness desired in the ciphered output of any proposed cipher. In fact, using TESTU01 and Practrand will save us time to run all the implemented tests within these tools. To prove that Speck-R acts well under these tools and possesses a high level of randomness in the ciphered image, we implement both Speck and Speck-R and adapt their code into these two tools.

6.6.1/ PROPOSED SCENARIO:

Using the ”testingRNG” [371] project released by Daniel Lemire which basically aims at testing popular random-number generators, randomness tests were conducted. In fact, the seed utilized is generated using the ”splitmix” [323] pseudo-random generator that generates 64 bits. Then, considering the worst case scenario, the plaintext is set and fixed to zeroes, and the nonce is changed by incrementing it by 1 after each iteration. This can be summarized as fixing the main key, round keys, plaintext, except for the nonce which is incremented by 1 from one iteration (Block) to another. In this strategy, and if the randomness tests are passed, we can say that this cipher possesses a high level of randomness even after changing only one bit in the nonce. The codes were implemented in C language, and for 4 TB of data for Practrand. The C codes for the Algorithms for both Speck and Speck-R are provided in Annex B.

After running these C codes in Practrand and TestU01, both of the ciphers, Speck and Speck-R succeeded the tests and no failure has been noted. Therefore, we showed by the aid of these tools, that the proposed cipher possesses a high level of randomness, with lower number of rounds and a high security level. In the following section, excessive tests are exerted to prove that Speck-R is also useful to be used for ciphering images which holds many intrinsic properties.

6.7/ SECURITY ANALYSIS

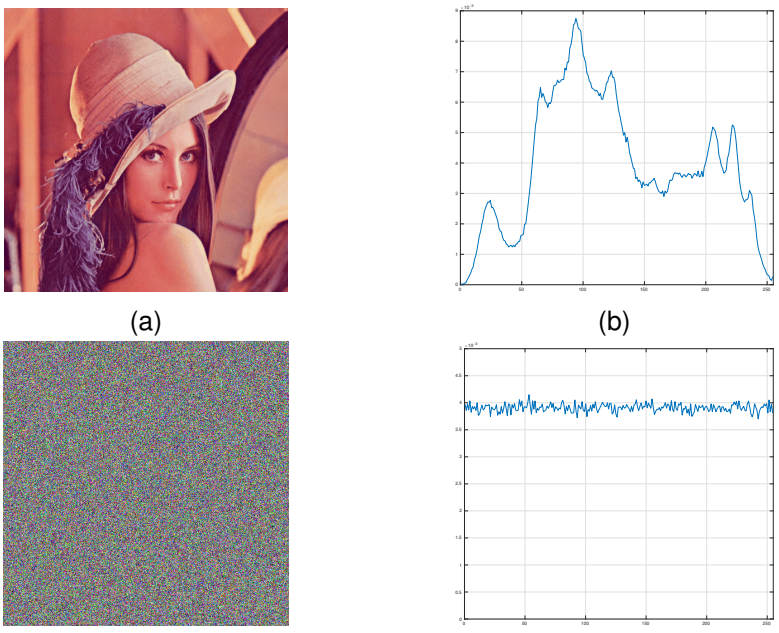
In this section, a security analysis is performed to validate the robustness of the proposed dynamic Speck-R. In fact, these tests, show that Speck-R can be also used for encrypting images that possess different intrinsic features and data is highly correlated. These tests show its immunity against different confidentiality attacks such as statistical, differential, and brute force attacks [238]. To prove that Speck-R is efficient for encrypting images, several tests were conducted.

6.7.1/ STATISTICAL ANALYSIS

A cipher scheme requires specific random properties in order to resist efficiently statistical attacks [191]. To prove the effectiveness of the proposed model, several statistical security tests were carried out to validate the uniformity and the independence properties. These tests are (1) **Uniformity Analysis**, (2) **Entropy test**, and the (3) **Correlation test.**

6.7.1.1/ UNIFORMITY ANALYSIS

To show the Uniformity of the ciphered image, two parameters were used: (a) Probability Density Function (PDF) analysis, and the (b) Chi-square test. First, the encrypted image should possess certain random properties to resist the common statistical attacks. The most commonly used property is the PDF of the encrypted image that should be uniform. This requires each symbol to have a probability close to $\frac{1}{n}$, where n is the number of symbols. This value means that there is significantly no clue to employ any statistical attack. The PDF of the original plain-image and its corresponding cipher-image are both



(a) (b)

(c) (d)

Figure 6.8: (a) Original Lenna, (b) PDF of original Lenna with size $512 \times 512 \times 3$, (c) Encrypted Lenna using Speck-R, (d) PDF of encrypted Lenna.

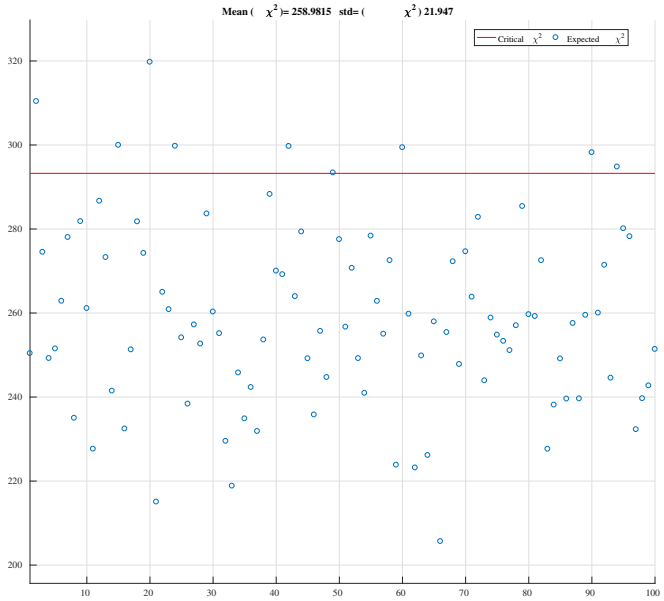


Figure 6.9: The Chi-Square test for the encrypted Lenna image using 100 different dynamic keys.

shown in Figure 6.8. It is clear that the PDF of the ciphered Speck-R image is close to 0.0039 ($\frac{1}{256}$). Additionally, in order to compute the level of uniformity of each encrypted image, the Chi-square test is applied according to equation 6.10:

$$\chi^2_{test} = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i} \tag{6.10}$$

k represents the number of gray levels (here we work in grey scale images, then k=256),

and o_i and e_i are the observed and expected occurrence frequencies of each gray level. This test aims at comparing the observed data with what we expect according to a specific hypothesis. Therefore, null hypothesis are formulated which are then rejected or retained with the help of statistical tests. The "significant level" is the probability value below which the null hypothesis is rejected, it can be also called the alpha level. According to [195], it is conventional to consider the null hypothesis false if the probability value is less than 0.05. In fact, having a significance level of 0.05 makes researchers 95% confident that the results represent a non-chance finding [160]. In addition, with a significance level of 0.05 and 256 number of intervals, the chi-square reaches a maximal value 293 [329]. So, all values lower than this value are acceptable and indicate the uniformity distribution of the histogram. This criterion is verified, by testing the chi-square for the Lenna image under 100 different dynamic keys. We can say that the redundancy of the plain image is hidden and does not provide any clue for applying statistical attack. In Figure 6.9, it can be seen that mean of the chi-square value for 100 iterations mean chi-square value for 100 iterations of encrypted Lena image is approximately equal to $258.9815 \leq 293$, which confirms the uniformity property of the encrypted image under the proposed algorithm.

6.7.1.2/ ENTROPY TEST

The information entropy of an image, M, is a parameter that measures the level of uncertainty in a random variable [260], and it is defined using the following equation:

$$H(m) = -\sum_{i=1}^{n} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{6.11}$$

$$H(m) = -\sum_{i=1}^{h^2} \frac{1}{h^2} \log_2 \frac{1}{h^2} = \log_2(h^2) \tag{6.12}$$

where $p(m_i)$ represents the occurrence probability of the symbol m_i and n is the total number of states of the information source. Note that the entropy is expressed in bits. The proposed entropy test measures the entropy at the sub-matrix level, where each sub-matrix has a size equal to h^2 bytes. This permits to quantify the uniformity at the sub-matrix level and not on the whole image. Each block can be considered as a truly random source with uniform distribution if it has an entropy equal or close to $\log_2(h^2)$. It is shown that the encrypted blocks always have an entropy close to the desired value 6 ($log_2(8{\times}8) = log_2(2^6) = 6$) in case $h = 8$. According to this, the proposed cipher ensures the uniformity and eliminates the redundancy between adjacent pixels. The Entropy analysis of original and encrypted Lena images under the use of a random dynamic key for $h = 8$ is shown in Figure 6.10. The results indicate that the encrypted sub-matrices have always an entropy close to the desired value 6. This result proves that the proposed cipher ensures uniformity and eliminates any redundancy between adjacent sub-matrices.

6.7.1.3/ TEST CORRELATION BETWEEN ORIGINAL AND CIPHER IMAGES

The high linear correlation among original image pixels must be removed to resist statistical attacks. Removing spatial redundancy will certainly result in an efficient cipher scheme [184, 318]. Having a correlation coefficient close to zero means that the cipher

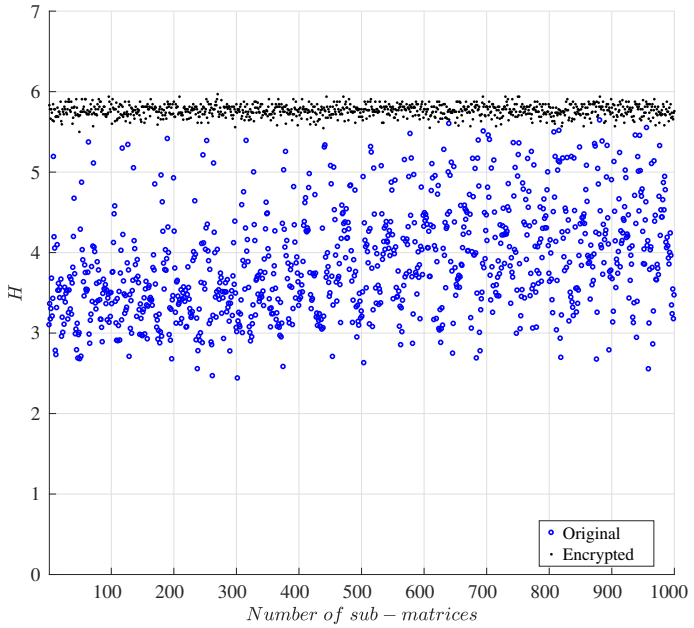


Figure 6.10: The Entropy analysis for the sub-matrices of encrypted Lena image under the use of Speck-R with a random dynamic key for $h = 8$.

scheme exhibits a high degree of randomness. The correlation test is performed by taking randomly $N = 4,066$ pairs of adjacent pixels from the known Lenna plain image and their corresponding cipher image. The correlation is done in horizontal, vertical and diagonal directions. The correlation coefficient r_{xy} is calculated using the following equations:

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x) \times D(y)}} \tag{6.13}$$

where

$$E_x = \frac{1}{N} \times \sum_{i=1}^{N} x_i$$

$$D_x = \frac{1}{N} \times \sum_{i=1}^{N} (x_i - E(x))^2$$

$$\text{cov}(x, y) = \frac{1}{N} \times \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y))$$

Obviously, the correlation between adjacent pixels in the plain image is high and its corresponding correlation coefficient is close to 1. Whereas, the correlation in the ciphered imaged is close to 0. Figure 6.11 shows the correlation between adjacent pixels in the different directions for a random secret key for the original and ciphered Lenna image, which clearly shows that the proposed scheme drastically reduces the spatial redundancy.

Moreover, for 16 iterations, the mean of the correlation in its three directions was calculated and it is clearly shown that the scattering effect of Speck-R removes any spatial correlation in the ciphered image. In Figure 6.12, it is clear that the mean is always close

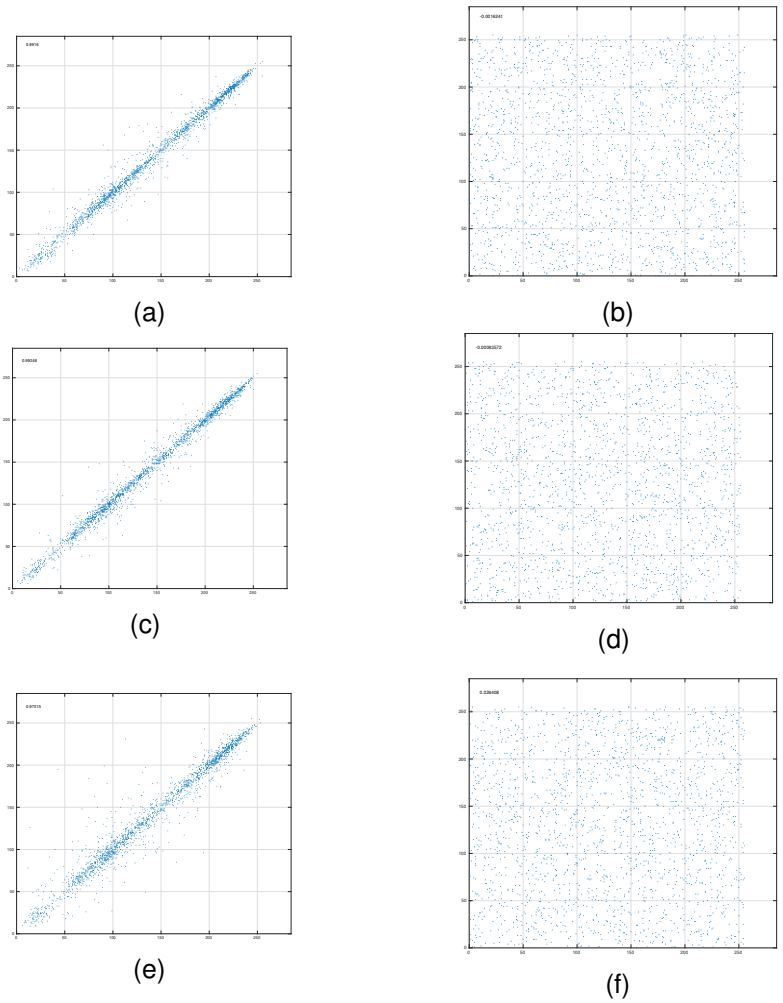


Figure 6.11: Correlation in adjacent pixels in original Lenna: (a) horizontally, (c) vertically and (e) diagonally and the correlation in adjacent pixels in ciphered Lenna: (b) horizontally, (d) vertically and (f) diagonally.

to zero which validates our proposal and renders this cipher immune against statistical attacks.

6.7.2/ VISUAL DEGRADATION

The degradation of the original image must be verified, in a way that the visual content of the ciphered image is not recognized. For this aspect, two well-known parameters are well known to measure the encryption visual quality and these are the Peak Signal-to-Noise Ratio (PSNR) [172] and the Structural Similarity Index (SSIM) [112].

PSNR is derived from the Mean Squared Error (MSE), which represents the cumulative squared error between an original and encrypted image. A low PSNR value demonstrates that a high difference between the original and the cipher image exists.
SSIM index [86] is defined after the Human Visual System (HVS), which has evolved, so that we can extract the structural information from the scene. Thus, the perceived quality of the image by the human eye is highly dependent on the loss of structural information

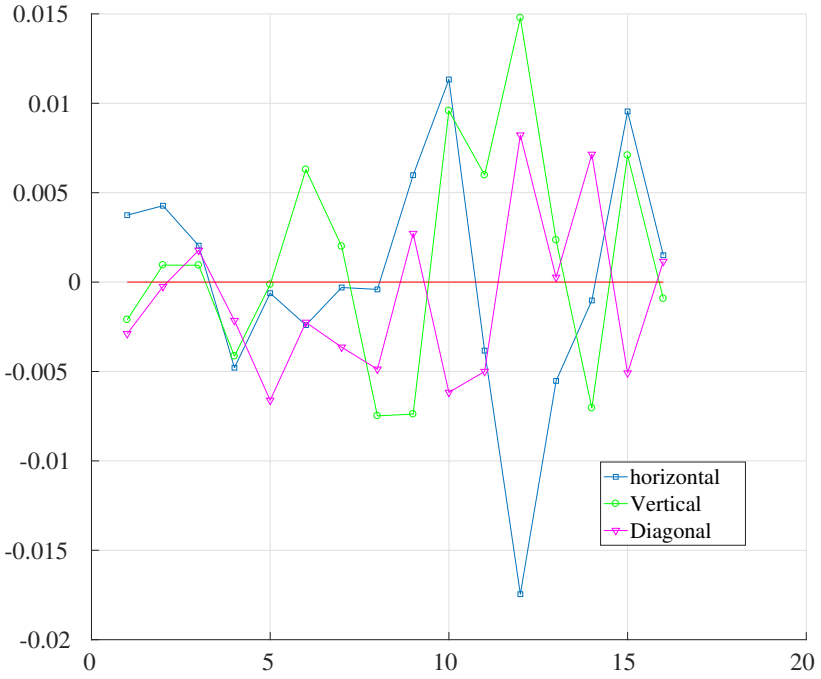


Figure 6.12: The mean of the correlation of the encrypted Lenna after 16 iterations.

in the image. The SSIM value lies in the interval [0, 1]. A value of 0 means that there is no correlation between the original and the cipher image, while a value close to 1 means that both images are approximately the same.

In this context, PSNR and SSIM were measured between the original and the encrypted Lena image for $1,000$ dynamic keys and presented in Figure 6.13. As shown, the mean PSNR value is 8.62 dB. This low value confirms that the proposed encryption technique provides a high difference between the original and the encrypted images. Also, the SSIM value did not exceed 0.011, which means that a high and adequate visual distortion is achieved using the proposed encryption process.
As a conclusion, the proposed cipher scheme has a sufficient visual degradation where no useful information or any clear pattern about the original image is revealed from the encrypted image.

6.7.2.1/ Difference Between plain And Cipher Image

Another criteria to measure the visual degradation is measuring the difference between original and encrypted images at the bit level. This value must reach a value very close to the ideal one (50%). In Figure 6.14, the difference between the original and cipher Lenna images for 1000 random dynamic keys is shown. The results show that the percentage difference is always close to 50%. Hence, the proposed cipher satisfies the independence criteria.

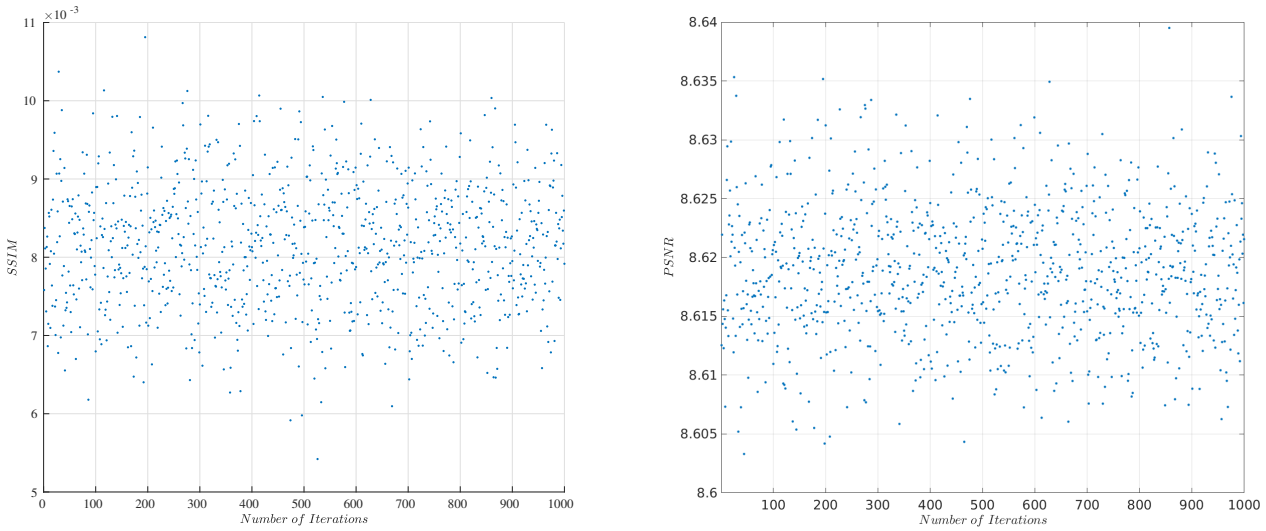


Figure 6.13: PSNR and SSIM variation between the original and the encrypted Lena image versus $1,000$ dynamic keys.

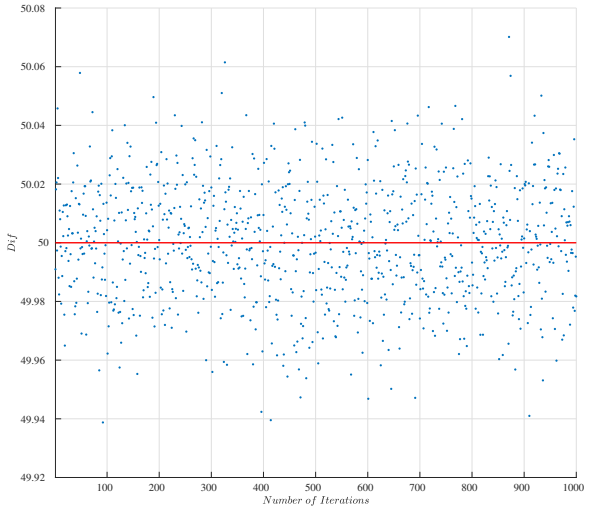


Figure 6.14: Percentage difference between plain and ciphered Lena for 1000 random dynamic keys.

6.8/ PERFORMANCE ANALYSIS

In this section, the performance of the proposed Speck-R is studied. In fact, the whole objective of this work is to increase the performance of the original Speck cipher, taking into account the high level of security. Looking forward to be implemented in IoT devices and small sensors, Speck-R undergoes different tests to prove its high performance in limited constrained devices. Two different aspects are studied, (1) error propagation and (2) the execution time required to fulfill the encryption process.

6.8.1/ PROPAGATION OF ERRORS

In this proposal, we chose to work with Speck-R in CTR mode (counter mode), since it does not suffer from error propagation and the image will be resilient to different kind of noises in the transmitting channel. This criterion should be as low as possible, which means that the error should not propagate to the whole transmitted image. Mainly, channel interference and noise in transmission are the main causes for any errors. A bit error means toggling the '0' bit into '1' and vice versa. In the proposed cipher, if the block is affected it will only affect the bit in the exact position of the ciphered image. It will not propagate to the neighboring blocks, and this is the cost of not insuring the avalanche effect in the whole image. In Figure 6.16, we show the encrypted image of Lenna Figure 6.15 after toggling the Least Significant Byte (LSB) in half of the blocks in the ciphered image, then we decrypted it in Figure 6.17. The results show that the decrypted image is disturbed, but it is well recognized. This shows that the error is not propagated in the image, which validates that the error is limited to its own block.

Figure 6.15: Lenna $512 \times 512 \times 3$

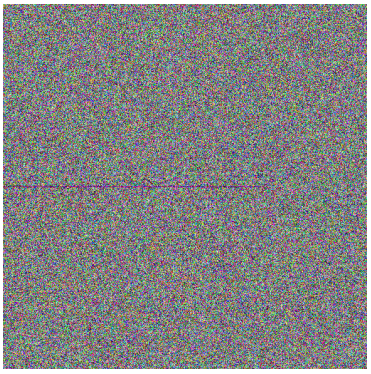


Figure 6.16: Encrypted Lenna after toggling.

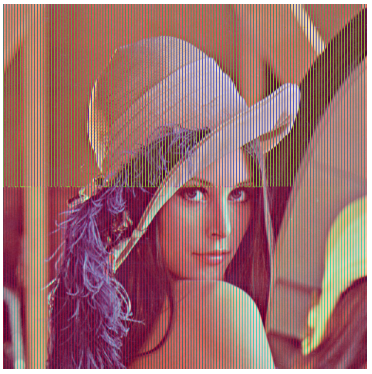


Figure 6.17: Dycrypted toggled Lenna.

6.8.2/ EXECUTION TIME

To validate that Speck-R is efficient for small limited devices, we tested the execution time on three different IoT devices: ATmega323p, Teensy 3.6 and DOIT ESP32. Below, Table 6.5 lists some of the specifications of these three IoT chips. The Figures 6.18 6.19 6.20 represents the three microchips used.

Device	ATmega328P	Teensy 3.6	DOIT ESP32
Flash Memory Size (KB)	32	256	4,096
Operating Voltage Range (V)	1.8 to 5.5	3.3V	3.3
Clock Speed (Mhz)	16	180	80
Processor	8 bit	32 bit	32 bit

Table 6.5: Specifications of ATmega323P,Teensy 3.6, and DOIT ESP32

Both algorithm, Speck and Speck-R were implemented on these three IoT microchips. The results are shown in Figures 6.21, 6.22 and 6.23. In fact, Figure 6.21 represents the time result in μsec of Speck and Speck-R when implemented on ATmega328P. It seems

Figure 6.18: ATmega328P

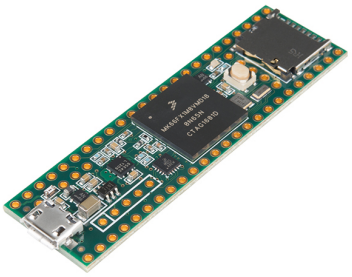


Figure 6.19: TEENSY 3.6

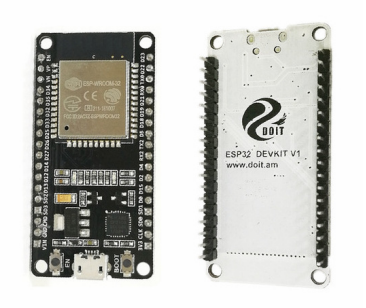


Figure 6.20: DOIT ESP32

that Speck-R performs better on this limited 8-bit micro controller chip. The encryption process is done for 16, 32, 64, 128, 256 and 512 bytes. For 16 bytes of data, the execution time for Speck is 940 μsec, while for Speck-R it is 304 μsec, which means that it took Speck-R less than half the time for Speck to encrypt 16 bytes. Then for 512 bytes, the execution times for Speck and Speck-R are 29668 μsec and 9840 μsec, respectively. It is clear that the time required for the encryption increases proportionally to the size of data. According to equation 6.14,the percentage of enhancement in the execution time is 66.8% when implementing both algorithms on ATmega328P.

$$Enhancement_{executiontime}\% = 100 \times (1 - \frac{Time_{Speck-R}}{Time_{Speck}}) \tag{6.14}$$

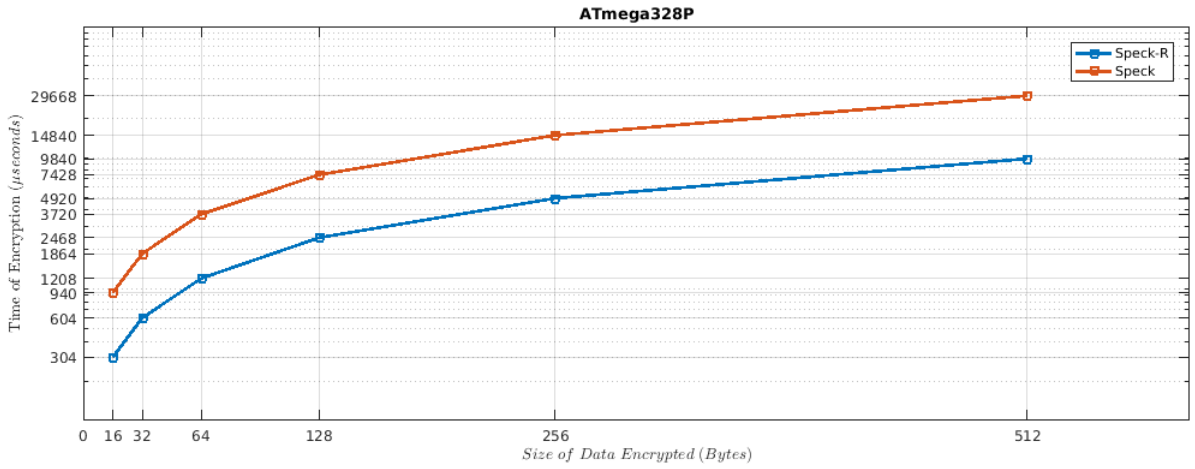


Figure 6.21: Execution time (μsec) versus the size of data (bytes) encrypted of Speck and Speck-R when implemented on ATmega328p.

Then, we implement both algorithms on the next IoT device, Teensy 3.6. An enhancement is recorded also for the sake of Speck-R. When encrypting 16 bytes, the two execution times were the same $2\mu sec$, but as the data gets larger, reaching 65536 bytes, the execution time of Speck-R is 5728 μsec, while for Speck 7015 μsec. The percentage of the enhancement is 18.34% when using Teensy 3.6.

The last microchip used is the DOIT ESP32. This chip which includes WiFi and Bluetooth, is widely used by researchers. Starting by 16 bytes of data to encrypt, Speck took 9

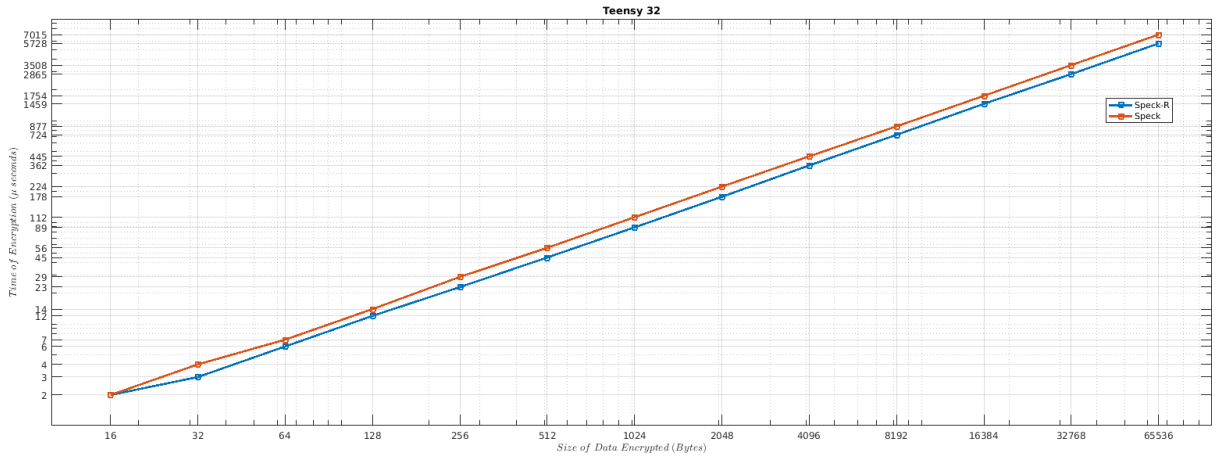


Figure 6.22: Execution time (μsec) versus the size of data (bytes) encrypted of Speck and Speck-R when implemented on Teensy 3.6.

μsec while Speck-R took 2 μsec. Then, reaching the maximum number of bytes, 4096 bytes, Speck-R with 328 μsec also possessed a higher performance comparing it to Speck having 597 μsec. In fact, the enhancement starts from 77% to reach a static 45% as the number of bytes increases.

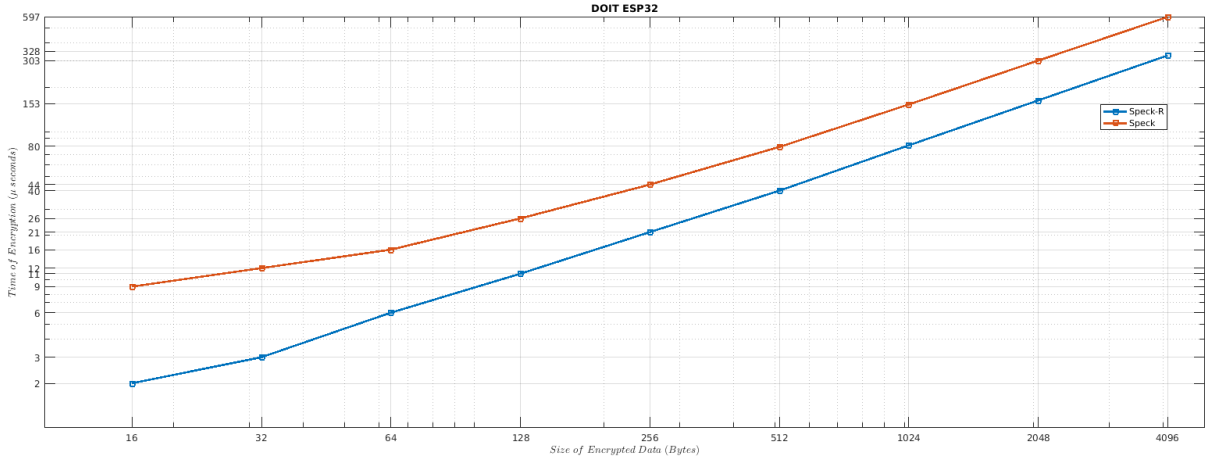


Figure 6.23: Execution time (μsec) versus the size of data (bytes) encrypted of Speck and Speck-R when implemented on DOIT ESP32.

As a conclusion, not only does Speck-R possess the security level necessary, but it also has a better execution time on these IoT chips which are limited in memory and in computational ability. Having simple operations such as Xor, shift, rotation and a simple Sbox, nominates this cipher to be a good proposal for today's security challenges.

6.9/ DISCUSSION AND CRYPT-ANALYSIS

To start by the quote of this chapter: **"Fully secure systems do not exist today and they will not exist in the future." — AdiShamir.** After all, researchers aim at enhancing as much as they can the level of security and strengthening the ciphers against well-known and new attacks. But, how can we know that the cipher is resilient against different kinds of attacks? First, the main two properties in this cipher are preserved which are confusion and diffusion. Confusion is preserved by using the proposed dynamic Sboxes and the diffusion is attained by using Xor, and shifting by α and β parameters.
Different statistical tests were performed and they proved that the proposed cipher satisfies the uniformity and independence properties. Hence, a high randomness level is achieved in a dynamic manner, which makes the proposed cipher immune against statistical attacks.
Using a dynamic key for every image proves that the cipher exhibits a high immunity against key-related attacks. Especially that the cryptographic parameters change as the dynamic key changes. Even if a cryptanalyst has a complete knowledge of the used primitives for a plain image, she/he will fail to extract information about the future plain images from the future cipher images, since they lack the dynamic key that is changed for every input image.
Also, the proposed cipher is immune to brute force attack since Speck it self can be used for different keys, starting from 64 reaching 256 bits. We chose to work on 96 bits, but this proposal can work on any other Speck version.
To sum up, dynamicity of the proposal and using different layers of Sbox adds more randomness and makes the proposed cipher scheme immune against the current and future powerful attacks such as chosen/known plain/cipher text attacks. In conclusion, the security level of the proposed cipher scheme is confirmed.

6.10/ CONCLUSION

In this chapter, we proposed a novel Speck which we call Speck-R. Speck-R comes to serve mostly the limited devices which are characterized by limited abilities and restricted power. Since we do not know how the future chips will operate, we intended to use simple operations that by convention will be supported by any old and new chip. Speck has been one of the most successful proposals in the domain of lightweight cryptography. Thus, based on Speck, we came out with the idea of a new Speck-R, which is a reduced version of Speck. To accomplish this goal, we added a confusion layer of substitution based on a dynamic approach. The Sboxes were built using a dynamic key and then changed according to the number of iterations. Adding dynamicity to the proposal also gave us a shield against different powerful attacks. We implemented Speck with 64 bits block, and 96 bits key using CTR mode. The achievement was reducing 26 rounds in Speck to 7 rounds in Speck-R and yet preserving a high level of security. Extensive tests were carried out to prove the robustness of this proposal. First, we started by validating that the $Sboxes$ used possess a high level of security and it was validated using our own written code, and by using the newly proposed tool SET. Then, following our proposal in Chapter 3, using Practrand, the randomness of the ciphered output was validated for 4 TB of processed data. Finally, to prove that this cipher works properly for images as well, since images are characterized by having a high correlation, many tests were exerted to show the reliability

for Speck-R to all kinds of data. The results supported the proposal and proved that this cipher has all the necessary aspects that nominate him as a successful cryptographical proposal in the field of limited and constrained devices. Moreover, and the most important criteria of Speck-R, is that it recorded at least 45% enhancement compared to Speck in terms of execution time. Both algorithms were implemented on small chips, that are mainly used for testing in IoT, and in the three cases Speck-R stepped ahead of Speck. All the results sustain the cipher and shows that all the important aspects that make it trustworthy are implanted within it.

IV

CONCLUSION

7

CONCLUSION AND PERSPECTIVES

7.1/ CONCLUSION

Being the core of digital safety, security is one of the most researched topics. Researchers have put a lot of effort in this field. However, due to the massive changes in the technological field, new terms and platforms have emerged. Internet of things, the big title, wraps many challenges that rise within it. The usage of embedded devices, small sensors and tiny devices that suffer from many constraints was the first obstacle faced. These devices suffer from limited memory, energy, life-time and processing power. Having a fraction of the total hardware of these devices dedicated to security mandated new security solutions to be proposed and adapted. There comes the concept of lightweight cryptography which is the focus of this dissertation.

In this manuscript, three contributions were done related to our field of interest, which is obtaining a better security and efficiency in terms of lightweight cryptography.

In chapter 2, we started by laying the foundations in the world of cryptography that can benefit any reader with a slightly low back ground in security. The different terms **symmetric, asymmetric** cryptography were explained and differentiated. Stating the security services that the cryptography offer, we then list the security primitives that offer them. Under the symmetric encryption, hash functions offers authentication and integrity, Message Authenticated Ciphers offer integrity and authentication as well. Then, the block and stream ciphers which operates differently but yet have the same objective to attain confidentiality. After that, authenticated ciphers are explained and how they offer both authentication and integrity. Then, under the title of asymmetric ciphers, public key cryptography, digital signatures and public key infrastructure are explained. We finally focus on how crucial it is to use both symmetric and asymmetric encryption when we want to reach all the security services. This is actually what is done in real systems, as the asymmetric encryption is used as a key wrapper to share the key for encryption/decryption processes which them selves will use a symmetric cipher. Such systems are called hybrid systems and they offer higher security and efficiency compared to other kinds of cryptographic systems.

In chapter 3, a state-of-the-art is represented. Regarding the field of lightweight cryptography, many works have been exerted in different perspectives. For example, some researchers focused on lightweight cryptography using block ciphers and others by using

stream ciphers. There have been competitions and evaluations to find the most suitable cipher for the special targeted audience, i.e the chips, sensors, embedded devices etc... In this chapter, we started first by explaining what is Internet of Things, the buzzword that is heard everywhere and anytime today. Then, one of the most used ciphers worldwide, AES, is also stated and we show why this cipher is not suitable for such systems. Therefore, we start by stating some of the famous ciphers which will be split into different directions. First, lightweight block ciphers are stated, followed by lightweight stream ciphers. We then list some of the dedicated authenticated encryption schemes that are used widely. Lightweight hash functions were also listed in the following part. Finally, we state some metrics that are used to classify whether the cipher is under the lightweight family or the ultra-lightweight one with a final touch of our personal cryptographic opinion.

In chapter 4, a survey on Vehicular Ad-hoc Network-VANET and the Internet of Vehicles-IoV is represented. The main objective behind this survey is to set a solid ground for researchers working in such fields. Knowing the challenges, the risks, the attacks and having a good classification will tend to have a better research and better results. We investigated one of the most challenging platforms that suffer from weak security solutions, as it has many obstacles in the way of good cryptography. We classified the attacks according to their impact on the layer they affect, and we surveyed most of the solutions for each attack. Moreover, attacks were classified according to their impact on the security service they target. We can say that these challenges, however, they are available in VANET, but many of them are available in many other platforms.

In chapter 5, we represented a new technique to be added to the existing testing techniques. As we know, there are plenty of tests that can be done for a cipher to prove that it possesses a good security level. However, in this proposed technique, a simple and yet a very efficient test can be done. TestU01 and Practrand, which are both famous tools for testing the randomness of the pseudo-random-generators, are proposed to test the output of the ciphers. These tools implement many important tests that can spare the researchers the effort of re-implementation. We benchmarked different ciphers that are used. Some of them were written by good programmers and others were implemented from trusted cryptographic libraries. Moreover, we intended to consider the worst case scenario, which is having the plaintext set as zeros, and then, after each round we changed only one bit in the key or in initial Vector (if used). The results showed the failure of some well-known ciphers either because of the code faults or because of a real problem in these ciphers. In short, we elect these methods as the easiest, and efficient test tools to use and we encourage other researchers to test their proposals before publishing their work.

In chapter 6 a new encryption algorithm is proposed based on Speck and named Speck-R. Knowing that Speck is proposed by NSA, we have the trust that its implementers have done most of the cyber-analysis. However, we propose a simpler Speck, named Speck-R, that reduces the number of rounds from 26 to 7. The ”R” stands for reduced after reducing the number of rounds in the original Speck. In this proposal, we added a dynamic key approach, which is represented by a dynamic substitution layer. The substitution layer is composed of three different Sboxes that will change according to a dynamically generated key. Three Sboxes will be initialized and the Sbox used in the encryption will change according to the size of the data being processed. Reaching a specific number of iterations, $Sbox_1$ will undergo substitution by $Sbox_2$ and then after another specific number of

iterations $Sbox_2$ will be substituted by $Sbox_3$. The proposed Speck-R has met the security perspectives we desired, which are attaining a high level of security and at the same time be as lightweight as possible. The randomness tests were done as explained in the previous chapter, and Speck-R passed them. Additionally, Speck-R has undergone different tests that proved it as a good candidate for the encryption of images as well. The reason for these tests is that images usually have more intrinsic features than texts and some specific tests prove the reliability of the security solution. Finally, to show that Speck-R is also a hardware dedicated cipher, as well as software a one, we tested it on three different IoT chips where the results back-boned our proposal. In the three cases, Speck-R stood ahead of Speck and the execution time was at least 18% reaching a maximum of 66.8% of enhancement in terms of execution time. As a conclusion, we nominate this cipher as a good competitor in the lightweight cryptography field.

7.2/ PERSPECTIVES

In this dissertation, we focused on explaining and proposing new techniques that can fit well in the lightweight cryptography field. In this part, we list our future works that can be added to improve the axis we work on and add more trust in the ciphers we submit.

Direction 1. National Institute of Standards and Technology has initiated a process to solicit, evaluate, and standardize lightweight cryptographic algorithms that are suitable for use in constrained environments where the performance of current NIST cryptographic standards is not acceptable. In September 2019 there were 32 finalists in Round 2. In the future work, we aim at benchmarking these proposed ciphers using the method we propose by TestU01 and Practrand. Additionally, we aim at implementing them on Contiki OS which is a an operating system for networked, memory-constrained systems with a focus on low-power wireless Internet of Things devices. Then, we aim at testing them in a real world condition using the IoT-LAB platform (https://www.iot-lab.info/).

Direction 2. Another proposal is called split-processing. Why should one sensor do all the job? If there are other sensors nor receiving/transmitting data, why jam one device with an amount of data that it is unable to process? In the future works, we will propose a secret sharing method that splits they encryption key and at the same time the data to be encrypted. The problem here is the synchronization among the nodes which we will try to manage a solution for. For example, in terms of surveillance, when there are more than one camera working together. There was a proposal that the cameras encrypt the data equally, which means to send parts of the moving image to the neighboring cameras to send and encrypt as well. Each camera will do a part of the of the job rather than making one camera watch, encrypt, send and process the images. Finding a way to share these keys secretly, synchronously and the data as well, in a lightweight manner is our next axes to work on.

Direction 3. In our work, the video content was not taken into consideration. Encrypting videos is one of the challenging aspects. A video is more expensive to encrypt/decrypt as it contains a very huge amount of data to be encrypted. Having a lightweight algorithm that can address videos as well is very important especially in VANET/IoV. We aim to test video encryption with a new or a previously proposed primitive. Then, we can also validate that this primitive is also able to be used for videos. First, we will try Speck-R to

see if it adapts well to videos and if it capable of processing the whole amount of data in a reasonable amount of time. If difficulties were encountered, we will propose another primitive and make sure that it suits videos as a first objective. Selective encryption in this case can be a solution, since we will select what exactly we want to encrypt (like avoid the repetition of data). Compressing the data before any encryption process is widely known as crypto-compression. Then, we will implement it on a small, limited device just as we did with Speck-R. Then, we will see if it fits the new environment or not and compare it to recently proposed algorithms dedicated for videos.

Direction 4. Finally, we speak about LoRaWAN that attracts many research works and is being implemented in different countries now. It is the heart of IoT projects, smart city, and industry and offers a cost-effective, high availability and scalability solutions. LoRaWAN is a MAC-layer protocol for long-range low-power communication. Since its release in 2015, it has experienced a rapid adoption in the field of Internet-of-Things (IoT). Yet, given that LoRaWAN is fairly novel, its level of security has not been thoroughly analyzed. For now, it uses AES encryption with 128 bit-keys. We intend to try and test new algorithms on the LoraWAN. This network has many advantages that can benefit the world of IoT, yet, there are many restrictions in the LoraWAN sensors especially in terms of the data rate (a maximum data rate of 27 kbps). We aim to test new efficient solutions that can be good candidate for LoraWAN platform. Having a cipher that can either take small block size of data and at the same time be resilient to different kinds of attacks will be a challenge. In our opinion, dynamicity is a solution for the limited abilities of these devices which can solve the replay and eavesdropping vulnerability for LoRaWaN as the key will change according to a specific time or session. Another vulnerability can be faced when introducing dynamicity is the avoiding the reuse of frame counter values, since in the proposed approach the Nonce will be generated by using another cipher primitive.

PERSONAL BIBLIOGRAPHY

PUBLISHED JOURNALS

- Noura Hassan, *Sleem Lama*, Noura Mohamad, Mansour Mohammad, Chehab Ali and Couturier Raphaël "A new efficient lightweight and secure image cipher scheme". **Multimedia Tools and Applications**, 77.12 (2018): 15457-15484.
- Noura Hassan, Chehab Ali, *Sleem Lama*, Noura Mohammad, Couturier Raphaël, Mansour, M. M. (2018). "One round cipher algorithm for multimedia IoT devices". **Multimedia tools and applications**, 77(14), 18383-18413.
- Noura Mohammad, Noura Hassan, Chehab Ali, Mansour Mouhamad, *Sleem Lama*, Couturier Raphaël (2018). "A dynamic approach for a lightweight and secure cipher for medical images". **Multimedia Tools and Applications**, 77(23), 31397-31426.

SUBMITTED JOURNALS

- Sleem Lama, Raphaël Couturier "TestU01 and Practrand: Tools for a Randomness Evaluation for Famous Multimedia Ciphers", **Multimedia Tools and Applications** Submitted: March, 2019.
- Sleem Lama, Raphaël Couturier "SPECK-R: AN ULTRA LIGHT-WEIGHT CRYPTOGRAPHIC SCHEME BASED ON SPECK", **Multimedia Tools and Applications** Submitted: November, 2019.

BIBLIOGRAPHY

[1] KERCKHOFFS, A. **La cryptographic militaire**. *Journal des sciences militaires* (1883), 5–38.

[2] ARTHUR, S. **Ciphering machine**, Jan. 24 1928. US Patent 1,657,411.

[3] KENDALL, M. G., AND SMITH, B. B. **Randomness and random sampling numbers**. *Journal of the royal Statistical Society 101*, 1 (1938), 147–166.

[4] SHANNON, C. E. **Communication theory of secrecy systems**. *Bell system technical journal 28*, 4 (1949), 656–715.

[5] DIFFIE, W., AND HELLMAN, M. **New directions in cryptography**. *IEEE transactions on Information Theory 22*, 6 (1976), 644–654.

[6] KOHNFELDER, L. M. **Towards a practical public-key cryptosystem.** PhD thesis, Massachusetts Institute of Technology, 1978.

[7] MERKLE, R., AND HELLMAN, M. **Hiding information and signatures in trapdoor knapsacks**. *IEEE transactions on Information Theory 24*, 5 (1978), 525–530.

[8] RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. **A method for obtaining digital signatures and public-key cryptosystems**. *Communications of the ACM 21*, 2 (1978), 120–126.

[9] MERKLE, R. **Secrecy, authentication, and public key systems**. *Ph. D. Thesis, Stanford University* (1979).

[10] RABIN, M. O. **Digitalized signatures and public-key functions as intractable as factorization**. Tech. rep., Massachusetts Inst of Tech Cambridge Lab for Computer Science, 1979.

[11] YUVAL, G. **How to swindle rabin**. *Cryptologia 3*, 3 (1979), 187–191.

[12] HELLMAN, M. E., AND MERKLE, R. C. **Public key cryptographic apparatus and method**, Aug. 19 1980. US Patent 4,218,582.

[13] MERKLE, R. C., AND HELLMAN, M. E. **On the security of multiple encryption**. *Communications of the ACM 24*, 7 (1981), 465–467.

[14] SHAMIR, A. **A polynomial time algorithm for breaking the basic merkle-hellman cryptosystem**. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)* (1982), IEEE, pp. 145–152.

[15] CHAUM, D. **Blind signatures for untraceable payments**. In *Advances in cryptology* (1983), Springer, pp. 199–203.

[16] ELGAMAL, T. **A public key cryptosystem and a signature scheme based on discrete logarithms**. *IEEE transactions on information theory 31*, 4 (1985), 469–472.

[17] FIAT, A., AND SHAMIR, A. **How to prove yourself: Practical solutions to identification and signature problems**. In *Conference on the Theory and Application of Cryptographic Techniques* (1986), Springer, pp. 186–194.

[18] WEBSTER, A. F., AND TAVARES, S. E. **On the design of s-boxes**. In *Advances in Cryptology* (Berlin, Heidelberg, 1986), CRYPTO '85, Springer-Verlag, pp. 523–534.

[19] DIFFIE, W. **The first ten years of public-key cryptography**. *Proceedings of the IEEE 76*, 5 (1988), 560–577.

[20] GOLDWASSER, S., MICALI, S., AND RACKOFF, C. **The knowledge complexity of interactive proof systems**. *SIAM Journal on computing 18*, 1 (1989), 186–208.

[21] MEIER, W., AND STAFFELBACH, O. **Nonlinearity criteria for cryptographic functions**. In *Workshop on the Theory and Application of of Cryptographic Techniques* (1989), Springer, pp. 549–562.

[22] ADAMS, C., AND TAVARES, S. **The structured design of cryptographically good s-boxes**. *Journal of Cryptology 3*, 1 (Jan 1990), 27–41.

[23] LAI, X., AND MASSEY, J. L. **A proposal for a new block encryption standard**. In *Workshop on the Theory and Application of of Cryptographic Techniques* (1990), Springer, pp. 389–404.

[24] NAOR, M., AND YUNG, M. **Public-key cryptosystems provably secure against chosen ciphertext attacks**. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing* (1990), Citeseer, pp. 427–437.

[25] PRENEEL, B., VAN LEEKWIJCK, W., VAN LINDEN, L., GOVAERTS, R., AND VANDEWALLE, J. **Propagation characteristics of boolean functions**. In *Workshop on the Theory and Application of of Cryptographic Techniques* (1990), Springer, pp. 161–173.

[26] BIHAM, E., AND SHAMIR, A. **Differential cryptanalysis of des-like cryptosystems**. *Journal of CRYPTOLOGY 4*, 1 (1991), 3–72.

[27] DING, C., XIAO, G., AND SHAN, W. **The stability theory of stream ciphers**, vol. 561. Springer Science & Business Media, 1991.

[28] RIVEST, R. **The md5 message-digest algorithm**.

[29] RIVEST, R. L. **The rc4 encryption algorithm. rsa data security**. *Inc., March 12* (1992), 9–2.

[30] ZHENG, Y., PIEPRZYK, J., AND SEBERRY, J. **Haval—a one-way hashing algorithm with variable length of output**. In *International workshop on the theory and application of cryptographic techniques* (1992), Springer, pp. 81–104.

[31] DAEMEN, J., GOVAERTS, R., AND VANDEWALLE, J. **A new approach to block cipher design**. In *International Workshop on Fast Software Encryption* (1993), Springer, pp. 18–32.

[32] MATSUI, M. **Linear cryptanalysis method for des cipher**. In *Workshop on the Theory and Application of of Cryptographic Techniques* (1993), Springer, pp. 386–397.

[33] SEBERRY, J., ZHANG, X.-M., AND ZHENG, Y. **Systematic generation of cryptographically robust s-boxes**. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (1993), ACM, pp. 171–182.

[34] BRANDS, S., AND CHAUM, D. **Distance-bounding protocols**. In *Advances in Cryptology—EUROCRYPT'93* (1994), Springer, pp. 344–359.

[35] COPPERSMITH, D. **The data encryption standard (des) and its strength against attacks**. *IBM journal of research and development 38*, 3 (1994), 243–250.

[36] RIVEST, R. L. **The rc5 encryption algorithm**. In *International Workshop on Fast Software Encryption* (1994), Springer, pp. 86–96.

[37] SCHNEIER, B. **Fast software encryption, cambridge security workshop proceedings**, 1994.

[38] WHEELER, D. J., AND NEEDHAM, R. M. **Tea, a tiny encryption algorithm**. In *International Workshop on Fast Software Encryption* (1994), Springer, pp. 363–366.

[39] DAEMEN, J. **Cipher and hash function design strategies based on linear and differential cryptanalysis**. PhD thesis, Doctoral Dissertation, March 1995, KU Leuven, 1995.

[40] KARN, P., METZGER, P., AND SIMPSON, W. **The esp triple des transform**. Tech. rep., 1995.

[41] ROGAWAY, P. **Problems with proposed ip cryptography**. *Unpublished manuscript* (1995).

[42] BELLOVIN, S. M. **Problem areas for the ip security protocols.** In *USENIX Security Symposium* (1996).

[43] BRUWER, F. J., SMIT, W., AND KUHN, G. J. **Microchips and remote control devices comprising same**, May 14 1996. US Patent 5,517,187.

[44] DOBBERTIN, H., BOSSELAERS, A., AND PRENEEL, B. **Ripemd-160: A strengthened version of ripemd**. In *International Workshop on Fast Software Encryption* (1996), Springer, pp. 71–82.

[45] JAKOBSSON, M., SAKO, K., AND IMPAGLIAZZO, R. **Designated verifier proofs and their applications**. In *International Conference on the Theory and Applications of Cryptographic Techniques* (1996), Springer, pp. 143–154.

[46] KAHN, D. **The Codebreakers: The comprehensive history of secret communication from ancient times to the internet**. Simon and Schuster, 1996.

[47] KATZ, J., MENEZES, A. J., VAN OORSCHOT, P. C., AND VANSTONE, S. A. **Handbook of applied cryptography**. CRC press, 1996.

[48] KNUDSEN, L. R., AND MEIER, W. **Improved differential attacks on rc5**. In *Annual International Cryptology Conference* (1996), Springer, pp. 216–228.

[49] DAEMEN, J., KNUDSEN, L., AND RIJMEN, V. **The block cipher square**. In *International Workshop on Fast Software Encryption* (1997), Springer, pp. 149–165.

[50] KELSEY, J., SCHNEIER, B., AND WAGNER, D. **Related-key cryptanalysis of 3-way, biham-des, cast, des-x, newdes, rc2, and tea**. In *International Conference on Information and Communications Security* (1997), Springer, pp. 233–246.

[51] KRAWCZYK, H., BELLARE, M., AND CANETTI, R. **Rfc 2104: Hmac: Keyed-hashing for message authentication**. *Internet Engineering Task Force 252* (1997).

[52] KRAWCZYK, H., CANETTI, R., AND BELLARE, M. **Hmac: Keyed-hashing for message authentication**.

[53] MATSUI, M. **New block encryption algorithm misty**. In *International Workshop on Fast Software Encryption* (1997), Springer, pp. 54–68.

[54] ATKINSON, R., AND KENT, S. **Ip encapsulating security payload (esp)**.

[55] CRACKING, D. **Secrets of encryption research**. *Wiretap Politics, and Chip Design, Electronic Frontier Foundation* (1998).

[56] DAEMEN, J., AND RIJMEN, V. **The block cipher bksq**. In *International Conference on Smart Card Research and Advanced Applications* (1998), Springer, pp. 236–245.

[57] HANDSCHUH, H., AND HEYS, H. M. **A timing attack on rc5**. In *International Workshop on Selected Areas in Cryptography* (1998), Springer, pp. 306–318.

[58] MARSAGLIA, G. **Diehard test suite**. *Online: http://www. stat. fsu. edu/pub/diehard 8*, 01 (1998), 2014.

[59] SCHNEIER, B., KELSEY, J., WHITING, D., WAGNER, D., HALL, C., AND FERGUSON, N. **Twofish: A 128-bit block cipher**. *NIST AES Proposal 15* (1998), 23.

[60] SELÇUK, A. A. **New results in linear cryptanalysis of rc5**. In *International Workshop on Fast Software Encryption* (1998), Springer, pp. 1–16.

[61] WHEELER, D. J., AND NEEDHAM, R. M. **Correction to xtea**. *Unpublished manuscript, Computer Laboratory, Cambridge University, England* (1998).

[62] EISLER, C. G., AND ENGSTROM, G. E. **Method and system for managing color specification using attachable palettes and palettes that refer to other palettes**, Dec. 28 1999. US Patent 6,008,816.

[63] KAUKONEN, K., AND THAYER, R. **A stream cipher encryption algorithm "arcfour"**, 1999.

[64] LIM, C. H. **A revised version of crypton: Crypton v1. 0**. In *International Workshop on Fast Software Encryption* (1999), Springer, pp. 31–45.

[65] OF STANDARDS, U. D. O. C. I., AND TECHNOLOGY. **Data encryption standard.federal information processing standards publication (fips)**, 1999. [Online; 1999].

[66] PUB, F. **Data encryption standard (des)**. *FIPS PUB* (1999), 46–3.

[67] SINGH, S. **The code book**, vol. 7. Doubleday New York, 1999.

[68] WEISER, M. **The computer for the 21st century, sigmobile mob**. *Comput. Commun. Rev 3*, 3 (1999), 3–11.

[69] AOKI, K., ICHIKAWA, T., KANDA, M., MATSUI, M., MORIAI, S., NAKAJIMA, J., AND TOKITA, T. **Camellia: A 128-bit block cipher suitable for multiple platforms—design andanalysis**. In *International Workshop on Selected Areas in Cryptography* (2000), Springer, pp. 39–56.

[70] BABBAGE, S., AND FRISCH, L. **On misty1 higher order differential cryptanalysis**. In *International Conference on Information Security and Cryptology* (2000), Springer, pp. 22–36.

[71] BARRETO, P., AND RIJMEN, V. **The khazad legacy-level block cipher**. *Primitive submitted to NESSIE 97* (2000), 106.

[72] KRUH, L. **Codes, ciphers & other cryptic & clandestine communication: Making and breaking secret messages from hieroglyphs to the internet**. *Cryptologia 24*, 1 (2000), 71.

[73] MARTI, S., GIULI, T. J., LAI, K., AND BAKER, M. **Mitigating routing misbehavior in mobile ad hoc networks**. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (2000), ACM, pp. 255–265.

[74] OHTA, H., AND MATSUI, M. **Rfc 2994, a description of the misty1 encryption algorithm**, 2000.

[75] BORISOV, N., GOLDBERG, I., AND WAGNER, D. **Intercepting mobile communications: the insecurity of 802.11**. In *Proceedings of the 7th annual international conference on Mobile computing and networking* (2001), ACM, pp. 180–189.

[76] DWORKIN, M. **Recommendation for block cipher modes of operation. methods and techniques**. Tech. rep., National Inst of Standards and Technology Gaithersburg MD Computer security Div, 2001.

[77] KRAWCZYK, H. **The order of encryption and authentication for protecting communications (or: How secure is ssl?)**. In *Annual International Cryptology Conference* (2001), Springer, pp. 310–331.

[78] MALYSHEV, D. S. **ebooks security – theory and practice**, 2001. [Online; 2001].

[79] MENEZES, A. J. **Paul c. van oorschot andscott a**. *Vanstone Handbook of Applied Cryptographyby* (2001).

[80] RIVEST, R. L., SHAMIR, A., AND TAUMAN, Y. **How to leak a secret**. In *International Conference on the Theory and Application of Cryptology and Information Security* (2001), Springer, pp. 552–565.

[81] BOND, P. J. **The keyed-hash message authentication code (hmac)**. *Federal Information Processing Standards Publication, FIPS PUB 198* (2002), 1–21.

[82] DIMITROVA, N., MCGEE, T., AND AGNIHOTRI, L. **Automatic signature-based spotting, learning and extracting of commercials and other video content**, Oct. 22 2002. US Patent 6,469,749.

[83] DOUCEUR, J. R. **The sybil attack**. In *Peer-to-peer Systems*. Springer, 2002, pp. 251–260.

[84] KLEIST, V. **The code book: the science of secrecy from ancient egypt to quantum cryptography [book review]**. *IEEE Annals of the History of Computing 24*, 2 (2002), 97–98.

[85] LEEN, G., AND HEFFERNAN, D. **Expanding automotive electronic systems**. *Computer 35*, 1 (2002), 88–93.

[86] LI, S., AND ZHENG, X. **Cryptanalysis of a chaotic image encryption method**. In *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on* (2002), vol. 2, IEEE, pp. II–708.

[87] PRENEEL, B. **New european schemes for signature, integrity and encryption (nessie): A status report**. In *International Workshop on Public Key Cryptography* (2002), Springer, pp. 297–309.

[88] STAJANO, F. **Security for ubiquitous computing**, vol. 1. Wiley Online Library, 2002.

[89] ZAPATA, M. G. **Secure ad hoc on-demand distance vector routing**. *ACM SIGMOBILE Mobile Computing and Communications Review 6*, 3 (2002), 106–107.

[90] ANDEM, V. R. **A cryptanalysis of the tiny encryption algorithm**. PhD thesis, University of Alabama, 2003.

[91] BOESGAARD, M., VESTERAGER, M., PEDERSEN, T., CHRISTIANSEN, J., AND SCAVENIUS, O. **Rabbit: A new high-performance stream cipher**. In *International Workshop on Fast Software Encryption* (2003), Springer, pp. 307–329.

[92] BURG, A. **Ad hoc network specific attacks**. In *Seminar Ad hoc networking: Concepts, Applications, and Security. Technische Universitat Munchen,'03* (2003).

[93] HABIB, A., HEFEEDA, M., AND BHARGAVA, B. K. **Detecting service violations and dos attacks.** In *NDSS* (2003).

[94] HAN, Z., FENG, W. X., HUI, L. Z., DA HAI, L., AND CHOU, L. Y. **A new image encryption algorithm based on chaos system**. In *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003* (2003), vol. 2, IEEE, pp. 778–782.

[95] HUANG, D., SINHA, A., AND MEDHI, D. **A double authentication scheme to detect impersonation attack in link state routing protocols**. In *IEEE International Conference on Communications, 2003. ICC'03.* (2003), vol. 3, IEEE, pp. 1723–1727.

[96] HUSSAIN, A., HEIDEMANN, J., AND PAPADOPOULOS, C. **A framework for classifying denial of service attacks**. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications* (2003), ACM, pp. 99–110.

[97] JONSSON, J., AND KALISKI, B. **Public-key cryptography standards (pkcs)# 1: Rsa cryptography specifications version 2.1**.

[98] RAFAELI, S., AND HUTCHISON, D. **A survey of key management for secure group communication**. *ACM Computing Surveys (CSUR) 35*, 3 (2003), 309–329.

[99] WARNER, J. S., AND JOHNSTON, R. G. **Gps spoofing countermeasures**. *Homeland Security Journal* (2003).

[100] WEISSTEIN, E. W. **Birthday problem**.

[101] BLUM, J., AND ESKANDARIAN, A. **The threat of intelligent collisions**. *IT Professional 6*, 1 (Jan 2004), 24–29.

[102] GOLLE, P., GREENE, D., AND STADDON, J. **Detecting and correcting malicious data in vanets**. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks* (2004), ACM, pp. 29–37.

[103] GUILLEY, S., HOOGVORST, P., AND PACALET, R. **Differential power analysis model and some results**. In *Smart Card Research and Advanced Applications Vi*. Springer, 2004, pp. 127–142.

[104] HUBAUX, J.-P., CAPKUN, S., AND LUO, J. **The security and privacy of smart vehicles**. *IEEE Security & Privacy Magazine 2*, LCA-ARTICLE-2004-007 (2004), 49–55.

[105] KILLOURHY, K. S., MAXION, R. A., AND TAN, K. M. **A defense-centric taxonomy based on attack manifestations**. In *Dependable Systems and Networks, 2004 International Conference on* (2004), IEEE, pp. 102–111.

[106] MIRKOVIC, J., AND REIHER, P. **A taxonomy of ddos attack and ddos defense mechanisms**. *ACM SIGCOMM Computer Communication Review 34*, 2 (2004), 39–53.

[107] PIRES JR, W. R., DE PAULA FIGUEIREDO, T. H., WONG, H. C., AND LOUREIRO, A. A. F. **Malicious node detection in wireless sensor networks**. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International* (2004), IEEE, p. 24.

[108] ROGAWAY, P. **Nonce-based symmetric encryption**. In *International Workshop on Fast Software Encryption* (2004), Springer, pp. 348–358.

[109] ROGAWAY, P., AND SHRIMPTON, T. **Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance**. In *International workshop on fast software encryption* (2004), Springer, pp. 371–388.

[110] STANDAERT, F.-X., PIRET, G., ROUVROY, G., QUISQUATER, J.-J., AND LEGAT, J.-D. **Iceberg: An involutional cipher efficient for block encryption in reconfigurable hardware**. In *International Workshop on Fast Software Encryption* (2004), Springer, pp. 279–298.

[111] STRAND, L. K. **Adaptive distributed firewall using intrusion detection**. Master's thesis, 2004.

[112] WANG, Z., BOVIK, A. C., SHEIKH, H. R., AND SIMONCELLI, E. P. **Image quality assessment: from error visibility to structural similarity**. *Image Processing, IEEE Transactions on 13*, 4 (2004), 600–612.

[113] WILLE, C. **Storing passwords-done right**. *last updated 1* (2004).

[114] BONO, S., GREEN, M., STUBBLEFIELD, A., JUELS, A., RUBIN, A. D., AND SZYDLO, M. **Security analysis of a cryptographically-enabled rfid device.** In *USENIX Security Symposium* (2005), vol. 31, pp. 1–16.

[115] CARLET, C. **On highly nonlinear s-boxes and their inability to thwart dpa attacks**. In *International Conference on Cryptology in India* (2005), Springer, pp. 49–62.

[116] DALAI, D. K., GUPTA, K. C., AND MAITRA, S. **Results on algebraic immunity for cryptographically significant boolean functions**. In *Progress in Cryptology - INDOCRYPT 2004* (Berlin, Heidelberg, 2005), A. Canteaut and K. Viswanathan, Eds., Springer Berlin Heidelberg, pp. 92–106.

[117] DOTZER, F., FISCHER, L., AND MAGIERA, P. **Vars: A vehicle ad-hoc network reputation system**. In *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a* (2005), IEEE, pp. 454–456.

[118] GILBERT, H., AND HANDSCHUH, H. **Fast Software Encryption (12 conf.)**. Springer, 2005.

[119] GLIGOR, V. D. **Light-weight cryptography—how light is light? keynote presentation at the information security summer school, florida state university. slide deck**, 2005.

[120] KOCH, W., AND SCHULTE, M. **The libgcrypt reference manual**. *Free Software Foundation Inc* (2005), 1–47.

[121] LIM, C. H., AND KORKISHKO, T. **mcrypton–a lightweight block cipher for security of low-cost rfid tags and sensors**. In *International Workshop on Information Security Applications* (2005), Springer, pp. 243–258.

[122] PARNO, B., AND PERRIG, A. **Challenges in securing vehicular networks**. In *Workshop on hot topics in networks (HotNets-IV)* (2005), pp. 1–6.

[123] SINGELEE, D., AND PRENEEL, B. **Location verification using secure distance bounding protocols**. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on* (2005), IEEE, pp. 7–pp.

[124] BRAEKEN, A. **Cryptographic properties of Boolean functions and S-boxes**. PhD thesis, phd thesis-2006, 2006.

[125] COMMITTEE SCC32. **IEEE P1609.4 standard for wireless access in vehicular environments (WAVE) - multi-channel operation**.

[126] DE CANNIÈRE, C. **Trivium: A stream cipher construction inspired by block cipher design principles**. In *International Conference on Information Security* (2006), Springer, pp. 171–186.

[127] HONG, D., SUNG, J., HONG, S., LIM, J., LEE, S., KOO, B.-S., LEE, C., CHANG, D., LEE, J., JEONG, K., AND OTHERS. **Hight: A new block cipher suitable for low-resource device**. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2006), Springer, pp. 46–59.

[128] KAMAT, P., BALIGA, A., AND TRAPPE, W. **An identity-based security framework for vanets**. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks* (2006), ACM, pp. 94–95.

[129] KOO, B. W., JANG, H. S., AND SONG, J. H. **On constructing of a 32× 32 binary matrix as a diffusion layer for a 256-bit block cipher**. In *International Conference on Information Security and Cryptology* (2006), Springer, pp. 51–64.

[130] KUMAR, S., PAAR, C., PELZL, J., PFEIFFER, G., AND SCHIMMLER, M. **Breaking ciphers with copacobana–a cost-optimized parallel code breaker**. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2006), Springer, pp. 101–118.

[131] LI, X., KANG, H., HARRINGTON, P., AND THOMAS, J. **Autonomic and trusted computing paradigms**. In *Autonomic and Trusted Computing*. Springer, 2006, pp. 143–152.

[132] NGAI, E. C., LIU, J., AND LYU, M. R. **On the intruder detection for sinkhole attack in wireless sensor networks**. In *2006 IEEE International Conference on Communications* (2006), vol. 8, IEEE, pp. 3383–3389.

[133] RAYA, M., PAPADIMITRATOS, P., AND HUBAUX, J.-P. **Securing vehicular communications**. *IEEE wireless communications 13*, 5 (2006), 8–15.

[134] RAYA, M., PAPADIMITRATOS, P., AND HUBAUX, J.-P. **Securing vehicular communications**. *Wireless Communications, IEEE 13*, 5 (October 2006), 8–15.

[135] STALLINGS, W. **The whirlpool secure hash function**. *Cryptologia 30*, 1 (2006), 55–67.

[136] STANDAERT, F.-X., PIRET, G., GERSHENFELD, N., AND QUISQUATER, J.-J. **Sea: A scalable encryption algorithm for small embedded applications**. In *International Conference on Smart Card Research and Advanced Applications* (2006), Springer, pp. 222–236.

[137] STEVENS, D. **Rot13 is used in windows? you're joking!**, 2006. [Online; 2006].

[138] VAUDENAY, S. **A classical introduction to cryptography: Applications for communications security**. Springer Science & Business Media, 2006.

[139] XIAO, B., YU, B., AND GAO, C. **Detection and localization of sybil nodes in vanets**. In *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks* (2006), ACM, pp. 1–8.

[140] YU, Y., YANG, Y., FAN, Y., AND MIN, H. **Security scheme for rfid tag**. *Auto-ID Labs Fudan University, White Paper* (2006).

[141] BOGDANOV, A., KNUDSEN, L. R., LEANDER, G., PAAR, C., POSCHMANN, A., ROBSHAW, M. J., SEURIN, Y., AND VIKKELSOE, C. **Present: An ultra-lightweight block cipher**. In *International workshop on cryptographic hardware and embedded systems* (2007), Springer, pp. 450–466.

[142] CHEN, L., ALMOUBAYED, K. A., AND LENEUTRE, J. **Detection and prevention of greedy behavior in ad hoc networks**. In *International Conference on Risks and Security of Internet and Systems (CRISIS 2007)* (2007).

[143] CHENG, L., HENTY, B. E., STANCIL, D. D., BAI, F., AND MUDALIGE, P. **Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 ghz dedicated short range communication (dsrc) frequency band**. *Selected Areas in Communications, IEEE Journal on 25*, 8 (2007), 1501–1516.

[144] DEGABRIELE, J. P., AND PATERSON, K. G. **Attacking the ipsec standards in encryption-only configurations**. In *2007 IEEE Symposium on Security and Privacy (SP'07)* (2007), IEEE, pp. 335–349.

[145] FREUDIGER, J., RAYA, M., FÉLEGYHÁZI, M., PAPADIMITRATOS, P., AND HUBAUX, J.-P. **Mix-zones for location privacy in vehicular networks**. In *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)* (2007), no. CONF.

[146] GUO, J., BAUGH, J. P., AND WANG, S. **A group signature based secure and privacy-preserving vehicular communication framework**. *Mobile Networking for Vehicular Environments 2007* (2007), 103–108.

[147] KARNADI, F. K., MO, Z. H., AND LAN, K.-C. **Rapid generation of realistic mobility models for vanet**. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE* (2007), IEEE, pp. 2506–2511.

[148] L'ECUYER, P., AND SIMARD, R. **Testu01: Ac library for empirical testing of random number generators**. *ACM Transactions on Mathematical Software (TOMS) 33*, 4 (2007), 22.

[149] LI, X., LI, M., SHU, W., AND WU, M. **A practical map-matching algorithm for gps-based vehicular networks in shanghai urban area**. In *Wireless, Mobile and Sensor Networks, 2007.(CCWMSN07). IET Conference on* (2007), IET, pp. 454–457.

[150] LO, N.-W., AND TSAI, H.-C. **Illusion attack on vanet applications-a message plausibility problem**. In *2007 IEEE Globecom Workshops* (2007), IEEE, pp. 1–8.

[151] PANAYAPPAN, R., TRIVEDI, J. M., STUDER, A., AND PERRIG, A. **Vanet-based approach for parking space availability**. In *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks* (2007), ACM, pp. 75–76.

[152] PAPADIMITRATOS, P., BUTTYAN, L., HUBAUX, J.-P., KARGL, F., KUNG, A., AND RAYA, M. **Architecture for secure and private vehicular communications**. In *Telecommunications, 2007. ITST'07. 7th International Conference on ITS* (2007), IEEE, pp. 1–6.

[153] PAPAPANAGIOTOU, K., MARIAS, G. F., AND GEORGIADIS, P. **A certificate validation protocol for vanets**. In *Globecom Workshops, 2007 IEEE* (2007), IEEE, pp. 1–9.

[154] RAYA, M., AND HUBAUX, J.-P. **Securing vehicular ad hoc networks**. *Journal of Computer Security 15*, 1 (2007), 39–68.

[155] SAMPIGETHAYA, K., LI, M., HUANG, L., AND POOVENDRAN, R. **Amoeba: Robust location privacy scheme for vanet**. *IEEE Journal on Selected Areas in communications 25*, 8 (2007), 1569–1589.

[156] SCHNEIER, B. **Applied cryptography: protocols, algorithms, and source code in C**. john wiley & sons, 2007.

[157] SHIRAI, T., SHIBUTANI, K., AKISHITA, T., MORIAI, S., AND IWATA, T. **The 128-bit blockcipher clefia**. In *International workshop on fast software encryption* (2007), Springer, pp. 181–195.

[158] SUN, J., ZHANG, C., AND FANG, Y. **An id-based framework achieving privacy and non-repudiation in vehicular ad hoc networks**. In *MILCOM 2007-IEEE Military Communications Conference* (2007), IEEE, pp. 1–7.

[159] TAMILSELVAN, L., AND SANKARANARAYANAN, D. V. **Prevention of impersonation attack in wireless mobile ad hoc networks**. *International Journal of Computer Science and Network Security (IJCSNS) 7*, 3 (2007), 118–123.

[160] VANVOORHIS, C. W., AND MORGAN, B. L. **Understanding power and rules of thumb for determining sample sizes**. *Tutorials in quantitative methods for psychology 3*, 2 (2007), 43–50.

[161] ZEGHID, M., MACHHOUT, M., KHRIJI, L., BAGANNE, A., AND TOURKI, R. **A modified aes based algorithm for image encryption**. *International Journal of Computer Science and Engineering 1*, 1 (2007), 70–75.

[162] ZHOU, T., CHOUDHURY, R. R., NING, P., AND CHAKRABARTY, K. **Privacy-preserving detection of sybil attacks in vehicular ad hoc networks**. In *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on* (2007), IEEE, pp. 1–8.

[163] ANDERSON, R. **Security engineering**. John Wiley & Sons, 2008.

[164] BABBAGE, S., AND DODD, M. **The mickey stream ciphers**. In *New Stream Cipher Designs*. Springer, 2008, pp. 191–209.

[165] BERNSTEIN, D. J. **Chacha, a variant of salsa20**. In *Workshop Record of SASC* (2008), vol. 8, pp. 3–5.

[166] BOUKERCHE, A., OLIVEIRA, H. A., NAKAMURA, E. F., AND LOUREIRO, A. A. **Vehicular ad hoc networks: A new challenge for localization-based systems**. *Computer communications 31*, 12 (2008), 2838–2849.

[167] DAN, T., AND XIAOJING, W. **Image encryption based on bivariate polynomials**. In *2008 international conference on computer science and software engineering* (2008), vol. 6, IEEE, pp. 193–196.

[168] EISENBARTH, T., KASPER, T., MORADI, A., PAAR, C., SALMASIZADEH, M., AND SHALMANI, M. T. M. **On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme**. In *Annual International Cryptology Conference* (2008), Springer, pp. 203–220.

[169] FAN, L., ZHOU, Y., AND FENG, D. **A fast implementation of computing the transparency order of s-boxes**. In *2008 The 9th International Conference for Young Computer Scientists* (2008), IEEE, pp. 206–211.

[170] GLASS, S., PORTMANN, M., AND MUTHUKKUMARASAMY, V. **Securing wireless mesh networks**. *Internet Computing, IEEE 12*, 4 (2008), 30–36.

[171] GUETTE, G., AND BRYCE, C. **Using tpms to secure vehicular ad-hoc networks (vanets)**. In *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*. Springer, 2008, pp. 106–116.

[172] HUYNH-THU, Q., AND GHANBARI, M. **Scope of validity of PSNR in image/video quality assessment**. *Electronics letters 44*, 13 (2008), 800–801.

[173] INDESTEEGE, S., KELLER, N., DUNKELMAN, O., BIHAM, E., AND PRENEEL, B. **A practical attack on keeloq**. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2008), Springer, pp. 1–18.

[174] JAKUBIAK, J., AND KOUCHERYAVY, Y. **State of the art and research challenges for vanets**. In *2008 5th IEEE Consumer Communications and Networking Conference* (2008), IEEE, pp. 912–916.

[175] JIANG, D., AND DELGROSSI, L. **Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments**. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE* (2008), IEEE, pp. 2036–2040.

[176] LANGLEY, C., LUCAS, R., AND FU, H. **Key management in vehicular ad-hoc networks**. In *2008 IEEE International Conference on Electro/Information Technology* (2008), IEEE, pp. 223–226.

[177] LEINMULLER, T., SCHMIDT, R., SCHOCH, E., HELD, A., AND SCHAFER, G. **Modeling roadside attacker behavior in vanets**. In *GLOBECOM Workshops, 2008 IEEE* (2008), IEEE, pp. 1–10.

[178] LI, C.-T., HWANG, M.-S., AND CHU, Y.-P. **A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks**. *Computer Communications 31*, 12 (2008), 2803–2814.

[179] LU, R., LIN, X., ZHU, H., HO, P.-H., AND SHEN, X. **Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications**. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (2008), IEEE.

[180] NAIT-ABDESSELAM, F., BENSAOU, B., AND TALEB, T. **Detecting and avoiding wormhole attacks in wireless ad hoc networks**. *IEEE Communications Magazine 46*, 4 (2008), 127–133.

[181] ONIEVA, J. A., SAUVERON, D., CHAUMETTE, S., GOLLMANN, D., AND MARKANTONAKIS, K. **Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks: Second IFIP WG 11.2 International Workshop, WISTP 2008, Seville, Spain, May 13-16, 2008**, vol. 5019. Springer, 2008.

[182] PLÖSSL, K., AND FEDERRATH, H. **A privacy aware and efficient security infrastructure for vehicular ad hoc networks**. *Computer Standards & Interfaces 30*, 6 (2008), 390–397.

[183] RAYA, M., PAPADIMITRATOS, P., GLIGOR, V. D., AND HUBAUX, J.-P. **On data-centric trust establishment in ephemeral ad hoc networks**. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (2008), IEEE.

[184] RHOUMA, R., AND BELGHITH, S. **Cryptanalysis of a new image encryption algorithm based on hyper-chaos**. *Physics Letters A 372*, 38 (2008), 5973–5978.

[185] SCHOCH, E., KARGL, F., WEBER, M., AND LEINMULLER, T. **Communication patterns in vanets**. *Communications Magazine, IEEE 46*, 11 (2008), 119–125.

[186] STEIN, W., AND OTHERS. **Sage: Open source mathematical software**. *7 December 2009* (2008).

[187] WALKER, J. **Ent: a pseudorandom number sequence test program**. *Software and documentation available at/www. fourmilab. ch/random/S* (2008).

[188] WANG, N.-W., HUANG, Y.-M., AND CHEN, W.-M. **A novel secure communication scheme in vehicular ad hoc networks**. *Computer communications 31*, 12 (2008), 2827–2837.

[189] WATANABE, D., IDEGUCHI, K., KITAHARA, J., MUTO, K., FURUICHI, H., AND KANEKO, T. **Enocoro-80: a hardware oriented stream cipher**. In *2008 Third International Conference on Availability, Reliability and Security* (2008), IEEE, pp. 1294–1300.

[190] WU, H. **The stream cipher hc-128**. In *New stream cipher designs*. Springer, 2008, pp. 39–47.

[191] XU, S., WANG, Y., WANG, J., AND TIAN, M. **Cryptanalysis of two chaotic image encryption schemes based on permutation and xor operations**. In *2008 International Conference on Computational Intelligence and Security* (2008), vol. 2, IEEE, pp. 433–437.

[192] YAN, G., OLARIU, S., AND WEIGLE, M. C. **Providing {VANET} security through active position detection**. *Computer Communications 31*, 12 (2008), 2883 – 2897. Mobility Protocols for ITS/VANET.

[193] ASHTON, K., AND OTHERS. **That 'internet of things' thing**. *RFID journal 22*, 7 (2009), 97–114.

[194] DOMINGO-FERRER, J., AND WU, Q. **Safety and privacy in vehicular communications**. In *Privacy in Location-Based Applications*. Springer, 2009, pp. 173–189.

[195] DU PREL, J.-B., HOMMEL, G., RÖHRIG, B., AND BLETTNER, M. **Confidence interval or p-value?: part 4 of a series on evaluation of scientific publications**. *Deutsches Ärzteblatt International 106*, 19 (2009), 335.

[196] HAMIEH, A., BEN-OTHMAN, J., AND MOKDAD, L. **Detection of radio interference attacks in vanet**. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE* (2009), IEEE, pp. 1–5.

[197] KOSCH, T., KULP, I., BECHLER, M., STRASSBERGER, M., WEYL, B., AND LASOWSKI, R. **Communication architecture for cooperative systems in europe**. *Communications Magazine, IEEE 47*, 5 (2009), 116–125.

[198] LO, N.-W., AND TSAI, H.-C. **A reputation system for traffic safety event on vehicular ad hoc networks**. *EURASIP Journal on Wireless Communications and Networking 2009* (2009), 9.

[199] MERRIAM-WEBSTER ONLINE. **Merriam-Webster Online Dictionary**, 2009.

[200] MOUSTAFA, H., AND ZHANG, Y. **Vehicular networks: techniques, standards, and applications**. Auerbach publications, 2009.

[201] PADMAVATHI, D. G., SHANMUGAPRIYA, M., AND OTHERS. **A survey of attacks, security mechanisms and challenges in wireless sensor networks**. *arXiv preprint arXiv:0909.0576* (2009).

[202] PARK, J. H. **Security analysis of mcrypton proper to low-cost ubiquitous computing devices and applications**. *International Journal of Communication Systems 22*, 8 (2009), 959–969.

[203] PARK, S., ASLAM, B., TURGUT, D., AND ZOU, C. C. **Defense against sybil attack in vehicular ad hoc network based on roadside unit support**. In *Military Communications Conference, 2009. MILCOM 2009. IEEE* (2009), IEEE, pp. 1–7.

[204] POSCHMANN, A. Y. **Lightweight cryptography: cryptographic engineering for a pervasive world**. In *PH. D. THESIS* (2009), Citeseer.

[205] SAFI, S. M., MOVAGHAR, A., AND MOHAMMADIZADEH, M. **A novel approach for avoiding wormhole attacks in vanet**. In *Internet, 2009. AH-ICI 2009. First Asian Himalayas International Conference on* (2009), IEEE, pp. 1–6.

[206] STUDER, A., BAI, F., BELLUR, B., AND PERRIG, A. **Flexible, extensible, and efficient vanet authentication**. *Communications and Networks, Journal of 11*, 6 (2009), 574–588.

[207] WANG, Y., AND LI, F. **Vehicular ad hoc networks**. In *Guide to wireless ad hoc networks*. Springer, 2009, pp. 503–525.

[208] WASEF, A., AND SHEN, X. **Edr: Efficient decentralized revocation protocol for vehicular ad hoc networks**. *Vehicular Technology, IEEE Transactions on 58*, 9 (Nov 2009), 5214–5224.

[209] WOLF, M. **Vehicular security mechanisms**. In *Security Engineering for Vehicular IT Systems*. Springer, 2009, pp. 121–165.

[210] **Intelligent transport systems — communications access for land mobiles (CALM) — architecture**. ISO 21217:2010, ISO TC204, Geneva, Switzerland, Apr. 2010.

[211] **Intelligent transport systems (its), security, threat, vulnerability and risk analysis (tvra)**. Tech. Rep. ETSI TR 102 893 V1.1.1, 03 2010.

[212] ALVAREZ-CUBERO, J. A., AND ZUFIRIA, P. J. **A c++ class for analysing vector boolean functions from a cryptographic perspective**. In *2010 International Conference on Security and Cryptography (SECRYPT)* (2010), IEEE, pp. 1–9.

[213] BAADACHE, A., AND BELMEHDI, A. **Avoiding black hole and cooperative black hole attacks in wireless ad hoc networks**. *arXiv preprint arXiv:1002.1681* (2010).

[214] BADEL, S., DAĞTEKIN, N., NAKAHARA, J., OUAFI, K., REFFÉ, N., SEPEHRDAD, P., SUŠIL, P., AND VAUDENAY, S. **Armadillo: a multi-purpose cryptographic primitive dedicated to hardware**. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2010), Springer, pp. 398–412.

[215] DHURANDHER, S. K., OBAIDAT, M. S., JAISWAL, A., TIWARI, A., AND TYAGI, A. **Securing vehicular networks: a reputation and plausibility checks-based approach**. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE* (2010), IEEE, pp. 1550–1554.

[216] DOTY-HUMPHREY, C. **Practically random: C++ library of statistical tests for rngs**. *URL: https://sourceforge. net/projects/pracrand* (2010).

[217] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **Intelligent Transport Systems (ITS); Communications Architecture**. EN 302 665 V1.1.1, ETSI, September 2010.

[218] FERGUSON, N., SCHNEIER, B., AND KOHNO, T. **Cryptography engineering**. *Design Princi* (2010).

[219] FUENTES, J. M. D., GONZÁLEZ-TABLAS, A. I., AND RIBAGORDA, A. **Overview of security issues in vehicular ad-hoc networks**.

[220] GRAFLING, S., MAHONEN, P., AND RIIHIJARVI, J. **Performance evaluation of ieee 1609 wave and ieee 802.11 p for vehicular communications**. In *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on* (2010), IEEE, pp. 344–348.

[221] GROVER, J., GAUR, M. S., AND LAXMI, V. **A novel defense mechanism against sybil attacks in vanet**. In *Proceedings of the 3rd international conference on Security of information and networks* (2010), ACM, pp. 249–255.

[222] HASBULLAH, H., SOOMRO, I. A., AND AB MANAN, J.-L. **Denial of Service (DOS) Attack and Its Possible Solutions in VANET**.

[223] ISAAC, J. T., ZEADALLY, S., AND CAMARA, J. S. **Security attacks and solutions for vehicular ad hoc networks**. *IET communications 4*, 7 (2010), 894–903.

[224] KNUDSEN, L., LEANDER, G., POSCHMANN, A., AND ROBSHAW, M. J. **Printcipher: a block cipher for ic-printing**. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2010), Springer, pp. 16–32.

[225] MINHAS, R., AND TILAL, M. **Effects of jamming on ieee 802.11 p systems**.

[226] PATHAN, A.-S. K. **Security of self-organizing networks: MANET, WSN, WMN, VANET**. CRC press, 2010.

[227] SAMARA, G., AL-SALIHY, W. A., AND SURES, R. **Security analysis of vehicular ad hoc nerworks (vanet)**. In *Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on* (2010), IEEE, pp. 55–60.

[228] SEPEHRDAD, P., VAUDENAY, S., AND VUAGNOUX, M. **Discovery and exploitation of new biases in rc4**. In *International Workshop on Selected Areas in Cryptography* (2010), Springer, pp. 74–91.

[229] SUN, J., ZHANG, C., ZHANG, Y., AND FANG, Y. **An identity-based security system for user privacy in vehicular ad hoc networks**. *Parallel and Distributed Systems, IEEE Transactions on 21*, 9 (2010), 1227–1239.

[230] WAGAN, A., MUGHAL, B., AND HASBULLAH, H. **Vanet security framework for trusted grouping using tpm hardware**. In *Communication Software and Networks, 2010. ICCSN '10. Second International Conference on* (Feb 2010), pp. 309–312.

[231] WAGAN, A. A., MUGHAL, B. M., AND HASBULLAH, H. **Vanet security framework for trusted grouping using tpm hardware: Group formation and message dissemination**. In *Information Technology (ITSim), 2010 International Symposium in* (2010), vol. 2, IEEE, pp. 607–611.

[232] WU, Q., DOMINGO-FERRER, J., AND GONZÁLEZ-NICOLÁS, U. **Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications**. *Vehicular Technology, IEEE Transactions on 59*, 2 (2010), 559–573.

[233] ZHANG, L., WU, Q., SOLANAS, A., AND DOMINGO-FERRER, J. **A scalable robust authentication protocol for secure vehicular communications**. *vehicular Technology, IEEE Transactions on 59*, 4 (2010), 1606–1617.

[234] **Intelligent transport systems (ITS); v2v application; part 1: Cooperative awareness application (caa) specification.**, 2011.

[235] AUMASSON, J.-P., NAYA-PLASENCIA, M., AND SAARINEN, M.-J. O. **Practical attack on 8 rounds of the lightweight block cipher klein**. In *International Conference on Cryptology in India* (2011), Springer, pp. 134–145.

[236] BOGDANOV, A., KNEŽEVIĆ, M., LEANDER, G., TOZ, D., VARICI, K., AND VERBAUWHEDE, I. **Spongent: A lightweight hash function**. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2011), Springer, pp. 312–325.

[237] CHIM, T. W., YIU, S.-M., HUI, L. C., AND LI, V. O. **Specs: Secure and privacy enhancing communications schemes for vanets**. *Ad Hoc Networks 9*, 2 (2011), 189–203.

[238] CHO, J.-S., YEO, S.-S., AND KIM, S. K. **Securing against brute-force attack: A hash-based rfid mutual authentication protocol using a secret value**. *Computer communications 34*, 3 (2011), 391–397.

[239] CRAMA, Y., AND HAMMER, P. L. **Boolean functions: Theory, algorithms, and applications**. Cambridge University Press, 2011.

[240] DAVID, M., RANASINGHE, D. C., AND LARSEN, T. **A2u2: a stream cipher for printed electronics rfid tags**. In *2011 IEEE International Conference on RFID* (2011), IEEE, pp. 176–183.

[241] ETSI TS 102 636-5-1. **Intelligent transport systems (ITS); vehicular communications; geonetworking; part 5: Transport protocols; sub-part 1: Basic transport protocol**. Version 1.1.1, Feb. 2011.

[242] GONG, Z., NIKOVA, S., AND LAW, Y. W. **Klein: a new family of lightweight block ciphers**. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues* (2011), Springer, pp. 1–18.

[243] GUO, J., PEYRIN, T., POSCHMANN, A., AND ROBSHAW, M. **The led block cipher**. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2011), Springer, pp. 326–341.

[244] KENNEY, J. B. **Dedicated short-range communications (dsrc) standards in the united states**. *Proceedings of the IEEE 99*, 7 (2011), 1162–1182.

[245] KNEZEVIC, M. **Efficient hardware implementations of cryptographic primitives (efficiënte hardware implementaties van cryptografische primitieven)**.

[246] LU, R., LIN, X., LUAN, T. H., LIANG, X., AND SHEN, X. **Pseudonym changing at social spots: An effective strategy for location privacy in vanets**. *IEEE transactions on vehicular technology 61*, 1 (2011), 86–96.

[247] MISHRA, B., NAYAK, P., BEHERA, S., AND JENA, D. **Security in vehicular adhoc networks: A survey**. In *Proceedings of the 2011 International Conference on Communication, Computing & Security* (New York, NY, USA, 2011), ICCCS '11, ACM, pp. 590–595.

[248] SABAHI, F. **The security of vehicular adhoc networks**. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on* (2011), IEEE, pp. 338–342.

[249] SHIBUTANI, K., ISOBE, T., HIWATARI, H., MITSUDA, A., AKISHITA, T., AND SHIRAI, T. **Piccolo: an ultra-lightweight blockcipher**. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2011), Springer, pp. 342–357.

[250] SUBRAMANYAN, B., CHHABRIA, V. M., AND BABU, T. S. **Image encryption based on aes key expansion**. In *2011 Second International Conference on Emerging Applications of Information Technology* (2011), IEEE, pp. 217–220.

[251] SUMRA, I. A., AB MANAN, J.-L., AND HASBULLAH, H. **Timing attack in vehicular network**. In *Proceedings of the 15th WSEAS International Conference on Computers, World Scientific and Engineering Academy and Society (WSEAS)* (2011), pp. 151–155.

[252] SUMRA, I. A., AHMAD, I., HASBULLAH, H., AND BIN AB MANAN, J.-L. **Classes of attacks in vanet**. In *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International* (2011), IEEE, pp. 1–5.

[253] SUMRA, I. A., AND HASBULLAH, H. **Trust and trusted computing in vanet**.

[254] SUMRA, I. A., HASBULLAH, H., AHMAD, I., AND BIN AB MANAN, J.-L. **Forming vehicular web of trust in vanet**. In *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International* (2011), IEEE, pp. 1–6.

[255] TURAN, M. S., BASSHAM, L. E., BURR, W., CHANG, D., ZHANG, S., DWORKIN, M. J., KELSEY, J. M., PAUL, S., PERALTA, R., PERLNER, R., AND OTHERS. **Status report on the second round of the SHA-3 cryptographic hash algorithm competition**. US Department of Commerce, National Institute of Standards and Technology, 2011.

[256] VIGHNESH, N., KAVITA, N., URS, S. R., AND SAMPALLI, S. **A novel sender authentication scheme based on hash chain for vehicular ad-hoc networks**. In *Wireless Technology and Applications (ISWTA), 2011 IEEE Symposium on* (2011), IEEE, pp. 96–101.

[257] WU, W., AND ZHANG, L. **Lblock: a lightweight block cipher**. In *International Conference on Applied Cryptography and Network Security* (2011), Springer, pp. 327–344.

[258] XIU-TAO, F. **Zuc algorithm: 3gpp lte international encryption standard [j]**. *Information Security and Communications Privacy 12* (2011).

[259] YEH, L.-Y., CHEN, Y.-C., AND HUANG, J.-L. **Paacp: A portable privacy-preserving authentication and access control protocol in vehicular ad hoc networks**. *Computer Communications 34*, 3 (2011), 447–456.

[260] ZHANG, G., AND LIU, Q. **A novel image encryption method based on total shuffling scheme**. *Optics Communications 284*, 12 (2011), 2775–2780.

[261] ZHANG, L., WU, Q., QIN, B., AND DOMINGO-FERRER, J. **Appa: aggregate privacy-preserving authentication in vehicular ad hoc networks**. In *Information Security*. Springer, 2011, pp. 293–308.

[262] **Status of its security standards**. Tech. Rep. Document HTG1-1, EU-US ITS Task Force Standards Harmonization Working Group Harmonization Task Group 1, Novembre 2012.

[263] ABDELRAHEEM, M. A. A. M. A., KNUDSEN, L. R., ZENNER, E., AND LEANDER, G. **Cryptanalysis of some lightweight symmetric ciphers**.

[264] AERTS, W., BIHAM, E., DE MOITIÚ, D., DE MULDER, E., DUNKELMAN, O., INDESTEEGE, S., KELLER, N., PRENEEL, B., VANDENBOSCH, G. A., AND VERBAUWHEDE, I. **A practical attack on keeloq**. *Journal of Cryptology 25*, 1 (2012), 136–157.

[265] AL-KAHTANI, M. **Survey on security attacks in vehicular ad hoc networks (vanets)**. In *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on* (Dec 2012), pp. 1–9.

[266] BIBHU, V., KUMAR, R., KUMAR, B. S., AND SINGH, D. K. **Performance analysis of black hole attack in vanet**. *International Journal Of Computer Network and Information Security 4*, 11 (2012), 47.

[267] BIRYUKOV, A., LEURENT, G., AND ROY, A. **Cryptanalysis of the "kindle" cipher**. In *International Conference on Selected Areas in Cryptography* (2012), Springer, pp. 86–103.

[268] BORGHOFF, J., CANTEAUT, A., GÜNEYSU, T., KAVUN, E. B., KNEZEVIC, M., KNUDSEN, L. R., LEANDER, G., NIKOV, V., PAAR, C., RECHBERGER, C., AND OTHERS. **Prince–a low-latency block cipher for pervasive computing applications**. In *International Conference on the Theory and Application of Cryptology and Information Security* (2012), Springer, pp. 208–225.

[269] HE, L., AND ZHU, W. T. **Mitigating dos attacks against signature-based authentication in vanets**. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on* (2012), vol. 3, IEEE, pp. 261–265.

[270] JEONG, K., KANG, H., LEE, C., SUNG, J., AND HONG, S. **Biclique cryptanalysis of lightweight block ciphers present, piccolo and led.** *IACR Cryptology ePrint Archive 2012* (2012), 621.

[271] JOVANOVIC, P., KREUZER, M., AND POLIAN, I. **An algebraic fault attack on the led block cipher.** *IACR Cryptology ePrint Archive 2012* (2012), 400.

[272] KAUSHIK, S. S. **Review of different approaches for privacy scheme in vanets**. *Int. J 5*, 2 (2012), 2231–1963.

[273] LAFITTE, F. **The boolfun package: Cryptographic properties of boolean functions**.

[274] LU, H., LI, J., AND GUIZANI, M. **A novel id-based authentication framework with adaptive privacy preservation for vanets**. In *Computing, Communications and Applications Conference (ComComAp), 2012* (Jan 2012), pp. 345–350.

[275] LUSHENG, M., KARIM, D., BAREND, J. V. W., AND YSKANDAR, H. **Evaluation and enhancement of ieee 802.11p standard: A survey**. vol. 1.

[276] MARTIN, K. M. **Everyday cryptography**. *The Australian Mathematical Society 231*, 6 (2012).

[277] MENDEL, F., RIJMEN, V., TOZ, D., AND VARICI, K. **Differential analysis of the led block cipher**. In *International Conference on the Theory and Application of Cryptology and Information Security* (2012), Springer, pp. 190–207.

[278] MOALLA, R., LABIOD, H., LONC, B., AND SIMONI, N. **Risk analysis study of its communication architecture**. In *Network of the Future (NOF), 2012 Third International Conference on the* (Nov 2012), pp. 1–5.

[279] NOGUEIRA, M., SILVA, H., SANTOS, A., AND PUJOLLE, G. **A security management architecture for supporting routing services on wanets**. *Network and Service Management, IEEE Transactions on 9*, 2 (2012), 156–168.

[280] RAWAT, A., SHARMA, S., AND SUSHIL, R. **Vanet: security attacks and its possible solutions**. *Journal of Information and Operations Management 3*, 1 (2012), 301–304.

[281] SUN, Y. **Linear cryptanalysis of light-weight block cipher iceberg**. In *Advances in Electronic Commerce, Web Application and Communication*. Springer, 2012, pp. 529–532.

[282] WANG, X., LIU, T., AND XIAO, G. **Certificate-based anonymous authentication protocol for vehicular ad-hoc network**. *IETE Technical Review 29*, 5 (2012), 388–393.

[283] WANG, X., TENG, L., AND QIN, X. **A novel colour image encryption algorithm based on chaos**. *Signal Processing 92*, 4 (2012), 1101–1108.

[284] WEI, Y.-C., AND CHEN, Y.-M. **An efficient trust management system for balancing the safety and location privacy in vanets**. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on* (June 2012), pp. 393–400.

[285] ZEADALLY, S., HUNT, R., CHEN, Y.-S., IRWIN, A., AND HASSAN, A. **Vehicular ad hoc networks (vanets): status, results, and challenges**. *Telecommunication Systems 50*, 4 (2012), 217–241.

[286] 186-1, F. I. P. S. P. **Digital signature standard (dss)**, 2013. [Online; 7/19/2013].

[287] ASLAM, B., WANG, P., AND ZOU, C. C. **Extension of internet access to vanet via satellite receive–only terminals**. *International Journal of Ad Hoc and Ubiquitous Computing 14*, 3 (2013), 172–190.

[288] AUMASSON, J.-P., HENZEN, L., MEIER, W., AND NAYA-PLASENCIA, M. **Quark: A lightweight hash**. *Journal of cryptology 26*, 2 (2013), 313–339.

[289] AUMASSON, J.-P., NEVES, S., WILCOX-O'HEARN, Z., AND WINNERLEIN, C. **Blake2: simpler, smaller, fast as md5**. In *International Conference on Applied Cryptography and Network Security* (2013), Springer, pp. 119–135.

[290] BEAULIEU, R., SHORS, D., SMITH, J., TREATMAN-CLARK, S., WEEKS, B., AND WINGERS, L. **The simon and speck families of lightweight block ciphers cryptology eprint archive**, 2013.

[291] CHUANG, M.-C., AND LEE, J.-F. **Team: Trust-extended authentication mechanism for vehicular ad hoc networks**. *IEEE systems journal 8*, 3 (2013), 749–758.

[292] DAEMEN, J., AND RIJMEN, V. **The design of Rijndael: AES-the advanced encryption standard**. Springer Science & Business Media, 2013.

[293] ELSA MATHEW, M., AND KUMAR, A. R. **Threat analysis and defence mechanisms in vanet**. *International Journal of Advanced Research in Computer Science and Software Engineering 3*, 1 (2013), 47–53.

[294] ENGOULOU, R. **Sécurisation des VANETS par la méthode de réputation des noeuds**. PhD thesis, École Polytechnique de Montréal, 2013.

[295] GILLANI, S., SHAHZAD, F., QAYYUM, A., AND MEHMOOD, R. **A survey on security in vehicular ad hoc networks**. In *Communication Technologies for Vehicles*, M. Berbineau, M. Jonsson, J.-M. Bonnin, S. Cherkaoui, M. Aguado, C. Rico-Garcia, H. Ghannoum, R. Mehmood, and A. Vinel, Eds., vol. 7865 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 59–74.

[296] HONG, D., LEE, J.-K., KIM, D.-C., KWON, D., RYU, K. H., AND LEE, D.-G. **Lea: A 128-bit block cipher for fast encryption on common processors**. In *International Workshop on Information Security Applications* (2013), Springer, pp. 3–27.

[297] JOHANSSON, T., AND NGUYEN, P. Q. **Advances in Cryptology–EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013, Proceedings**, vol. 7881. Springer, 2013.

[298] KLONOWSKI, M., AND KOZA, M. **Countermeasures against sybil attacks in wsn based on proofs-of-work**. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks* (2013), ACM, pp. 179–184.

[299] LIU, Q., WU, Q., AND YONG, L. **A hierarchical security architecture of vanet**.

[300] MALLA, A. M., AND SAHU, R. K. **Security attacks with an effective solution for dos attacks in vanet**. *International Journal of Computer Applications 66*, 22 (2013), 45–49.

[301] RAY, B., DOUGLAS, S., JASON, S., STEFAN, T., BRYAN, W., AND LOUIS, W. **The simon and speck families of lightweight block ciphers**. *Technical report, Cryptology ePrint Archive, Report./404* (2013).

[302] SAEED, I. A., SELAMAT, A., AND ABUAGOUB, A. M. **A survey on malware and malware detection systems**. *International Journal of Computer Applications 67*, 16 (2013).

[303] SONG, J., LEE, K., AND LEE, H. **Biclique cryptanalysis on lightweight block cipher: Hight and piccolo**. *International Journal of Computer Mathematics 90*, 12 (2013), 2564–2580.

[304] SONG, L., AND HU, L. **Differential fault attack on the prince block cipher**. In *International Workshop on Lightweight Cryptography for Security and Privacy* (2013), Springer, pp. 43–54.

[305] TEAM, R. C., AND OTHERS. **R: A language and environment for statistical computing**.

[306] VERMA, K., HASBULLAH, H., AND KUMAR, A. **Prevention of dos attacks in vanet**. *Wireless personal communications 73*, 1 (2013), 95–126.

[307] BOGDANOV, A., MENDEL, F., REGAZZONI, F., RIJMEN, V., AND TISCHHAUSER, E. **Ale: Aes-based lightweight authenticated encryption**. In *Fast Software Encryption* (Berlin, Heidelberg, 2014), S. Moriai, Ed., Springer Berlin Heidelberg, pp. 447–466.

[308] DAEINABI, A., AND RAHBAR, A. G. **An advanced security scheme based on clustering and key distribution in vehicular ad-hoc networks**. *Computers & Electrical Engineering 40*, 2 (2014), 517 – 529.

[309] DINUR, I. **Improved differential cryptanalysis of round-reduced speck**. In *International Conference on Selected Areas in Cryptography* (2014), Springer, pp. 147–164.

[310] ENGOULOU, R. G., BELLAÏCHE, M., PIERRE, S., AND QUINTERO, A. {**VANET**} **security surveys**. *Computer Communications 44*, 0 (2014), 1 – 13.

[311] KATZ, J., AND LINDELL, Y. **Introduction to modern cryptography**. Chapman and Hall/CRC, 2014.

[312] KORTESNIEMI, Y., AND SÄRELÄ, M. **Survey of certificate usage in distributed access control**. *Computers & Security 44* (2014), 16–32.

[313] KUMAR, N., IQBAL, R., MISRA, S., AND RODRIGUES, J. J. **An intelligent approach for building a secure decentralized public key infrastructure in** {**VANET**}. *Journal of Computer and System Sciences*, 0 (2014), –.

[314] LA, V. H., AND CAVALLI, A. R. **Security attacks and solutions in vehicular ad hoc networks: a survey**.

[315] LI, Z., AND CHIGAN, C. **On joint privacy and reputation assurance for vehicular ad hoc networks**. *Mobile Computing, IEEE Transactions on 13*, 10 (2014), 2334–2344.

[316] MEJRI, M. N., AND BEN-OTHMAN, J. **Detecting greedy behavior by linear regression and watchdog in vehicular ad hoc networks**. In *2014 IEEE Global Communications Conference* (2014), IEEE, pp. 5032–5037.

[317] MEJRI, M. N., BEN-OTHMAN, J., AND HAMDI, M. **Survey on** {**VANET**} **security challenges and possible cryptographic solutions**. *Vehicular Communications 1*, 2 (2014), 53 – 66.

[318] NOROUZI, B., SEYEDZADEH, S. M., MIRZAKUCHAKI, S., AND MOSAVI, M. R. **A novel image encryption based on hash function with only two-round diffusion process**. *Multimedia systems 20*, 1 (2014), 45–64.

[319] PARK, M., AND KIM, J. **Differential fault analysis of the block cipher lea**. *Journal of the Korea Institute of Information Security and Cryptology 24*, 6 (2014), 1117–1127.

[320] PIETRO, R. D., GUARINO, S., VERDE, N., AND DOMINGO-FERRER, J. **Security in wireless ad-hoc networks – a survey**. *Computer Communications 51*, 0 (2014), 1 – 20.

[321] SARKAR, P., AND IWATA, T. **Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, China, December 7-11, 2014**, vol. 8874. Springer, 2014.

[322] SELVAM, R., SHANMUGAM, D., AND ANNADURAI, S. **Side channel attacks: Vulnerability analysis of prince and rectangle using dpa.** *IACR Cryptology ePrint Archive 2014* (2014), 644.

[323] STEELE JR, G. L., LEA, D., AND FLOOD, C. H. **Fast splittable pseudorandom number generators**. In *ACM SIGPLAN Notices* (2014), vol. 49, ACM, pp. 453–472.

[324] WADI, S. M., AND ZAINAL, N. **High definition image encryption algorithm based on aes modification**. *Wireless personal communications 79*, 2 (2014), 811–829.

[325] ARMKNECHT, F., AND MIKHALEV, V. **On lightweight stream ciphers with shorter internal states**. In *International Workshop on Fast Software Encryption* (2015), Springer, pp. 451–470.

[326] BANIK, S., BOGDANOV, A., ISOBE, T., SHIBUTANI, K., HIWATARI, H., AKISHITA, T., AND REGAZZONI, F. **Midori: A block cipher for low energy**. In *International Conference on the Theory and Application of Cryptology and Information Security* (2015), Springer, pp. 411–436.

[327] BAYSAL, A., AND ŞAHIN, S. **Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors**. In *Lightweight Cryptography for Security and Privacy* (2015), Springer, pp. 58–76.

[328] BITAM, S., MELLOUK, A., AND ZEADALLY, S. **Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks**. *IEEE Wireless Communications 22*, 1 (2015), 96–102.

[329] CHEN, J.-X., ZHU, Z.-L., FU, C., ZHANG, L.-B., AND ZHANG, Y. **An efficient image encryption scheme using lookup table-based confusion and diffusion**. *Nonlinear Dynamics 81*, 3 (2015), 1151–1166.

[330] COX, J. **Why you don't roll your own crypto**, 2015. [Online; 2015].

[331] DINU, D., BIRYUKOV, A., GROSSSCHÄDL, J., KHOVRATOVICH, D., LE CORRE, Y., AND PERRIN, L. **Felics–fair evaluation of lightweight cryptographic systems**. In *NIST Workshop on Lightweight Cryptography* (2015), vol. 128.

[332] DWORKIN, M. J. **Sha-3 standard: Permutation-based hash and extendable-output functions**. Tech. rep., 2015.

[333] GUERRERO-IBANEZ, J. A., ZEADALLY, S., AND CONTRERAS-CASTILLO, J. **Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies**. *IEEE Wireless Communications 22*, 6 (2015), 122–128.

[334] GUPTA, A., AND KAUSHAL, R. **Improving spam detection in online social networks**. In *2015 International conference on cognitive computing and information processing (CCIP)* (2015), IEEE, pp. 1–6.

[335] MEULEN, R. **Gartner says by 2020, a quarter billion connected vehicles will enable new in-vehicle services and automated driving capabilities**. *Gartner, STAMFORD, Conn* (2015).

[336] PARK, J.-H., KIM, T.-J., AN, H.-J., WON, Y.-S., AND HAN, D.-G. **Side channel attacks on lea and its countermeasures**. *Journal of the Korea Institute of Information Security and Cryptology 25*, 2 (2015), 449–456.

[337] RABIEH, K., MAHMOUD, M. M., AZER, M., AND ALLAM, M. **A secure and privacy-preserving event reporting scheme for vehicular ad hoc networks**. *Security and Communication Networks 8*, 17 (2015), 3271–3281.

[338] SAINI, M., ALELAIWI, A., AND SADDIK, A. E. **How close are we to realizing a pragmatic vanet solution? a meta-survey**. *ACM Computing Surveys (CSUR) 48*, 2 (2015), 29.

[339] TURTON, W. **Cryptography expert casts doubt on encryption in isis' favorite messaging app**, 2015. [Online; 2015].

[340] VERDULT, R., GARCIA, F. D., AND EGE, B. **Dismantling megamos crypto: Wirelessly lockpicking a vehicle immobilizer**. In *Supplement to the Proceedings of 22nd {USENIX} Security Symposium (Supplement to {USENIX} Security 15)* (2015), pp. 703–718.

[341] ZHANG, W., BAO, Z., LIN, D., RIJMEN, V., YANG, B., AND VERBAUWHEDE, I. **Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms**. *Science China Information Sciences 58*, 12 (2015), 1–15.

[342] ABHIMANYU, K. **How connected cars are turning into revenue-generating machines**, 2016. [Online; 2016].

[343] BEIERLE, C., JEAN, J., KÖLBL, S., LEANDER, G., MORADI, A., PEYRIN, T., SASAKI, Y., SASDRICH, P., AND SIM, S. M. **The skinny family of block ciphers and its low-latency variant mantis**. In *Annual International Cryptology Conference* (2016), Springer, pp. 123–153.

[344] DINU, D., PERRIN, L., UDOVENKO, A., VELICHKOV, V., GROSSSCHÄDL, J., AND BIRYUKOV, A. **Design strategies for arx with provable bounds: Sparx and lax**. In *International Conference on the Theory and Application of Cryptology and Information Security* (2016), Springer, pp. 484–513.

[345] DOBRAUNIG, C., EICHLSEDER, M., MENDEL, F., AND SCHLÄFFER, M. **Ascon v1. 2**. *Submission to the CAESAR Competition* (2016).

[346] EAST, C. M., PROGRAM, N. A., AND PROGRAM, C. T. **Why telegram's security flaws may put iran's journalists at risk**, 2016. [Online; 2016].

[347] FAWAZ, Z., NOURA, H., AND MOSTEFAOUI, A. **An efficient and secure cipher scheme for images confidentiality preservation**. *Signal Processing: Image Communication 42* (2016), 90–108.

[348] GRIFFITHS, S. **Never run out of food again! smart mat warns you when you're low on milk while fridge cam lets you remotely check what you already have during your weekly shop**, 2016. [Online; 2016].

[349] JAKOBSEN, J., AND ORLANDI, C. **On the cca (in) security of mtproto.** In *SPSM@ CCS* (2016), pp. 113–116.

[350] TOUTOUH, J., AND ALBA, E. **Light commodity devices for building vehicular ad hoc networks: An experimental study**. *Ad Hoc Networks 37* (2016), 499–511.

[351] WOLFSSL USER MANUAL. **User Manual – Version 3.9.0, wolfSSL**, 2016. [Online; 2016].

[352] WU, H. **Acorn: a lightweight authenticated cipher (v3)**. *Candidate for the CAESAR Competition. See also https://competitions. cr. yp. to/round3/acornv3. pdf* (2016).

[353] YANG, Q., HU, L., SUN, S., AND SONG, L. **Extension of meet-in-the-middle technique for truncated differential and its application to roadrunner**. In *International Conference on Network and System Security* (2016), Springer, pp. 398–411.

[354] BARKER, E., AND MOUHA, N. **Recommendation for the triple data encryption algorithm (tdea) block cipher**. Tech. rep., National Institute of Standards and Technology, 2017.

[355] BEAULIEU, R., SHORS, D., SMITH, J., TREATMAN-CLARK, S., WEEKS, B., AND WINGERS, L. **Notes on the design and analysis of simon and speck.** *IACR Cryptology ePrint Archive 2017* (2017), 560.

[356] BIRYUKOV, A., AND PERRIN, L. P. **State of the art in lightweight symmetric cryptography**.

[357] CHERKAOUI, B., BENI-HSSANE, A., AND ERRITALI, M. **Quality control chart for detecting the black hole attack in vehicular ad-hoc networks**. *Procedia computer science 113* (2017), 170–177.

[358] CONTRERAS-CASTILLO, J., ZEADALLY, S., AND GUERRERO-IBAÑEZ, J. A. **Internet of vehicles: Architecture, protocols, and security**. *IEEE internet of things Journal 5*, 5 (2017), 3701–3709.

[359] EL-SAYED, H., SANKAR, S., PRASAD, M., PUTHAL, D., GUPTA, A., MOHANTY, M., AND LIN, C.-T. **Edge of things: The big picture on the integration of edge, iot and the cloud in a distributed computing environment**. *IEEE Access 6* (2017), 1706–1717.

[360] MANOGARAN, G., LOPEZ, D., THOTA, C., ABBAS, K. M., PYNE, S., AND SUNDARASEKAR, R. **Big data analytics in healthcare internet of things**. In *Innovative healthcare systems for the 21st century*. Springer, 2017, pp. 263–284.

[361] PALMER, D. **175,000 iot cameras can be remotely hacked thanks to flaw, says security researcher**, 2017. [Online; 2017].

[362] PALMER, D. **Security flaws in children's smartwatches make them vulnerable to hackers**, 2017. [Online; 2017].

[363] SARIBEKYAN, H., AND MARGVELASHVILI, A. **Security analysis of telegram**.

[364] SUN, S., GERAULT, D., LAFOURCADE, P., YANG, Q., TODO, Y., QIAO, K., AND HU, L. **Analysis of aes, skinny, and others with constraint programming**.

[365] TODO, Y. **Integral cryptanalysis on full misty1**. *Journal of Cryptology 30*, 3 (2017), 920–959.

[366] ABDI, L., AND MEDDEB, A. **In-vehicle augmented reality tsr to improve driving safety and enhance the driver's experience**. *Signal, Image and Video Processing 12*, 1 (2018), 75–82.

[367] HASSAN, A. N., KAIWARTYA, O., ABDULLAH, A. H., SHEET, D. K., AND RAW, R. S. **Inter vehicle distance based connectivity aware routing in vehicular adhoc networks**. *Wireless Personal Communications 98*, 1 (2018), 33–54.

[368] HATZIVASILIS, G., FYSARAKIS, K., PAPAEFSTATHIOU, I., AND MANIFAVAS, C. **A review of lightweight block ciphers**. *Journal of Cryptographic Engineering 8*, 2 (2018), 141–184.

[369] ISO/IEC. **Information technology — security techniques — lightweight cryptography — part 3: Stream ciphers**, 2018. [Online; 2018].

[370] KWASNICKI, M. **Strong encryption for small payloads on Arduino**, 2018. [Online; 2018].

[371] LEMIRE, D. **testingRNG**, 2018. [Online; 2018].

[372] LOUISE MATSAKIS, I. L. **Everything we know about facebook's massive security breach**, 2018. [Online; 2018].

[373] MILLMAN, R. **Connected cars report: 125 million vehicles by 2022, 5g coming**, 2018. [Online; 2018].

[374] NAITO, Y., MATSUI, M., SUGAWARA, T., AND SUZUKI, D. **Saeb: A lightweight blockcipher-based aead mode of operation**. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2018), 192–217.

[375] NOURA, H., CHEHAB, A., SLEEM, L., NOURA, M., COUTURIER, R., AND MANSOUR, M. M. **One round cipher algorithm for multimedia iot devices**. *Multimedia tools and applications 77*, 14 (2018), 18383–18413.

[376] NOURA, H., SLEEM, L., NOURA, M., MANSOUR, M. M., CHEHAB, A., AND COUTURIER, R. **A new efficient lightweight and secure image cipher scheme**. *Multimedia Tools and Applications 77*, 12 (2018), 15457–15484.

[377] O'BRIEN, B. **https://www.ariasystems.com/blog/will-profit-connected-cars/**, 2018. [Online; 2018].

[378] ORGANIZATION, W. H. **Global status report on road safety 2018**, 2018. [Online; 2018].

[379] FOUCHAL, H., BOURDY, E., WILHELM, G., AND AYAIDA, M. **Secured communications on vehicular networks over cellular networks**. In *International Conference on Distributed Computing and Internet Technology* (2019), Springer, pp. 31–41.

[380] IRWIN, L. **List of data breaches and cyber attacks in august 2019 – 114.6 million records leaked**, 2019. [Online; 2019].

[381] OSBORNE, C. **Whatsapp vulnerability exploited through malicious gifs to hijack chat sessions**, 2019. [Online; 2019].

[382] ÖZKAYNAK, F. **An analysis and generation toolbox for chaotic substitution boxes: a case study based on chaotic labyrinth rene thomas system**. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* (2019), 1–10.

[383] RAMESH, S. M., AND ALKHZAIMI, H. **Side channel analysis of sparx-64/128: Cryptanalysis and countermeasures**. In *International Conference on Cryptology in Africa* (2019), Springer, pp. 352–369.

[384] YIRKA, B. **Two major security vulnerabilities found in pdf files**, 2019. [Online; 2019].

[385] BIRYUKOV, A., AND PERRIN, L. **State of the art in lightweight symmetric cryptography**. *http://orbilu. uni. lu/handle/10993/31319, 24.05. 2018)*.

[386] DAEMEN, J., AND RIJMEN, V. **Aes proposal: Rijndael**.

[387] DHAMGAYE, A., AND CHAVHAN, N. **Survey on security challenges in vanet**.

[388] ETSI TS 101 539-2. **Intelligent transport system (its); v2v application; part 2: Intersection collision risk warning (icrw) application specification.**

[389] ETSI TS 101 539-3. **Intelligent transport system (its); v2v application; part 3: Longitudinal collision risk warning (lcrw) application specification**.

[390] EU-US ITS TASK FORCE 1-1. **Standards harmonization working group harmonization task group 1 , "current status of security standards"**.

[391] FIPS, P. **197, advanced encryption standard (aes), 2001. us department of commerce/national institute of standards and technology. true random number generator**. *Constructive Side-Channel Analysis and Secure Design 7275*, 151–166.

[392] IGURE, V., AND WILLIAMS, R. **Taxonomies of attacks and vulnerabilities in computer systems**. *Communications Surveys & Tutorials, IEEE 10*, 1, 6–19.

[393] KARAGIANNIS, G., ALTINTAS, O., EKICI, E., HEIJENK, G., JARUPAN, B., LIN, K., AND WEIL, T. **Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions**. *Communications Surveys & Tutorials, IEEE 13*, 4, 584–616.

[394] RAW, R. S., KUMAR, M., AND SINGH, N. **Security challenges, issues and their solutions for vanet**.

[395] SRIDEVI, B., AND GOPIKA, M. **Masquerade attack detection and prevention using enhanced key management techniques**.

[396] TOOR, Y., MUHLETHALER, P., AND LAOUITI, A. **Vehicle ad hoc networks: Applications and related technical issues**. *Communications Surveys & Tutorials, IEEE 10*, 3, 74–88.

V

ANNEXES

A

ANNEXE A

```
rng=RNG_stdin64, seed=0x5c742375
length= 4 gigabytes (2^32 bytes), time= 127 seconds
  Test Name                         Raw       Processed     Evaluation
  [Low4/64]BRank(12):2K(1)          R= +14.7  p~=  1.8e-5   unusual
  ...and 200 test result(s) without anomalies

rng=RNG_stdin64, seed=0x5c742375
length= 8 gigabytes (2^33 bytes), time= 251 seconds
  Test Name                         Raw       Processed     Evaluation
  BCFN(2+2,13-0,T)                  R=  -7.1  p =1-1.4e-3   unusual
  [Low4/64]BRank(12):2K(1)          R= +14.7  p~=  1.8e-5   unusual
  ...and 210 test result(s) without anomalies

rng=RNG_stdin64, seed=0x5c742375
length= 16 gigabytes (2^34 bytes), time= 497 seconds
  no anomalies in 223 test result(s)

rng=RNG_stdin64, seed=0x5c742375
length= 32 gigabytes (2^35 bytes), time= 982 seconds
  no anomalies in 233 test result(s)

rng=RNG_stdin64, seed=0x5c742375
length= 64 gigabytes (2^36 bytes), time= 1964 seconds
  no anomalies in 244 test result(s)

rng=RNG_stdin64, seed=0x5c742375
length= 128 gigabytes (2^37 bytes), time= 3902 seconds
  no anomalies in 255 test result(s)

rng=RNG_stdin64, seed=0x5c742375
length= 256 gigabytes (2^38 bytes), time= 7722 seconds
  Test Name                         Raw       Processed     Evaluation
  FPF-14+6/16:cross                 R=  +5.2  p =  9.1e-5   mildly suspicious
  ...and 264 test result(s) without anomalies

rng=RNG_stdin64, seed=0x5c742375
length= 512 gigabytes (2^39 bytes), time= 15424 seconds
  Test Name                         Raw       Processed     Evaluation
  FPF-14+6/16:cross                 R=  +9.3  p =  3.6e-8   very suspicious
  ...and 275 test result(s) without anomalies

rng=RNG_stdin64, seed=0x5c742375
length= 1 terabyte (2^40 bytes), time= 30363 seconds
  Test Name                         Raw       Processed     Evaluation
  FPF-14+6/16:cross                 R= +17.1  p =  1.0e-14    FAIL !
  ...and 286 test result(s) without anomalies

(END)
```

(a)

```
length= 4 gigabytes (2^32 bytes), time= 149 seconds


  no anomalies in 201 test result(s)

rng=RNG_stdin64, seed=0x42fa531
length= 8 gigabytes (2^33 bytes), time= 297 seconds
  no anomalies in 212 test result(s)

rng=RNG_stdin64, seed=0x42fa531
length= 16 gigabytes (2^34 bytes), time= 588 seconds
  no anomalies in 223 test result(s)

rng=RNG_stdin64, seed=0x42fa531
length= 32 gigabytes (2^35 bytes), time= 1165 seconds
  no anomalies in 233 test result(s)

rng=RNG_stdin64, seed=0x42fa531
length= 64 gigabytes (2^36 bytes), time= 2326 seconds
  Test Name                         Raw       Processed     Evaluation
  [Low1/64]FPF-14+6/16:all          R=  -4.5  p =1-5.5e-4   unusual
  ...and 243 test result(s) without anomalies

rng=RNG_stdin64, seed=0x42fa531
length= 128 gigabytes (2^37 bytes), time= 4648 seconds
  no anomalies in 255 test result(s)

rng=RNG_stdin64, seed=0x42fa531
length= 256 gigabytes (2^38 bytes), time= 9214 seconds
  Test Name                         Raw       Processed     Evaluation
  FPF-14+6/16:cross                 R=  +5.2  p =  9.1e-5   mildly suspicious
  ...and 264 test result(s) without anomalies

rng=RNG_stdin64, seed=0x42fa531
length= 512 gigabytes (2^39 bytes), time= 18408 seconds
  Test Name                         Raw       Processed     Evaluation
  FPF-14+6/16:cross                 R=  +9.3  p =  3.6e-8   very suspicious
  [Low16/64]FPF-14+6/16:cross       R=  +6.1  p =  1.6e-5   mildly suspicious
  ...and 274 test result(s) without anomalies

rng=RNG_stdin64, seed=0x42fa531
length= 1 terabyte (2^40 bytes), time= 36711 seconds
  Test Name                         Raw       Processed     Evaluation
  FPF-14+6/16:cross                 R= +17.1  p =  1.0e-14    FAIL !
  [Low16/64]FPF-14+6/16:cross       R=  +9.6  p =  1.8e-8   very suspicious
  ...and 285 test result(s) without anomalies

(END)
```

(b)

Figure A.1: (a) Practrand failure result of RC4 in Wolfcrypt library and (b) Libgcrypt library.

```
  [Low4/64]FPF-14+6/16:(16,14-5)    R= +23.6  p =  2.1e-19    FAIL !
  [Low4/64]FPF-14+6/16:(17,14-6)    R= +18.1  p =  6.7e-14    FAIL
  [Low4/64]FPF-14+6/16:(19,14-8)    R=  +9.4  p =  8.4e-7   mildly suspicious
  [Low4/64]FPF-14+6/16:(20,14-8)    R=  +8.3  p =  4.8e-6   unusual
  [Low4/64]FPF-14+6/16:all          R=+431.1  p =  4.5e-404   FAIL !!!!!!!
  [Low4/64]FPF-14+6/16:all2         R=+36285  p = 0           FAIL !!!!!!!!
  [Low4/64]FPF-14+6/16:cross        R=  +4.0  p =  1.0e-3   unusual
  [Low1/64]BCFN(2+0,13-0,T)         R= +92.4  p =  6.3e-49    FAIL !!!!
  [Low1/64]BCFN(2+1,13-0,T)         R= +92.8  p =  3.6e-49    FAIL !!!!
  [Low1/64]BCFN(2+2,13-0,T)         R= +95.2  p =  1.9e-50    FAIL !!!!
  [Low1/64]BCFN(2+3,13-0,T)         R= +91.3  p =  2.3e-48    FAIL !!!!
  [Low1/64]BCFN(2+4,13-1,T)         R= +65.3  p =  1.3e-34    FAIL !!!
  [Low1/64]BCFN(2+5,13-1,T)         R= +62.3  p =  5.5e-33    FAIL !!!
  [Low1/64]BCFN(2+6,13-2,T)         R= +56.7  p =  1.4e-28    FAIL !!
  [Low1/64]BCFN(2+7,13-3,T)         R= +31.4  p =  1.1e-14    FAIL
  [Low1/64]BCFN(2+8,13-3,T)         R= +39.0  p =  3.0e-18    FAIL !
  [Low1/64]BCFN(2+9,13-4,T)         R= +23.2  p =  4.3e-10   VERY SUSPICIOUS
  [Low1/64]BCFN(2+10,13-5,T)        R= +14.0  p =  1.1e-5   unusual
  [Low1/64]BCFN(2+11,13-5,T)        R= +14.9  p =  5.1e-6   unusual
  [Low1/64]BCFN(2+13,13-6,T)        R= +17.4  p =  2.1e-6   unusual
  [Low1/64]DC6-9x1Bytes-1           R=+475.6  p =  7.1e-224   FAIL !!!!!!
  [Low1/64]Gap-16:A                 R=+516.3  p =  1.3e-337   FAIL !!!!!!!
  [Low1/64]Gap-16:B                 R=+183.7  p =  5.5e-157   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(0,14-0)     R=+134.9  p =  2.0e-124   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(1,14-0)     R=+127.5  p =  1.3e-117   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(2,14-0)     R=+127.5  p =  1.2e-117   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(3,14-0)     R=+128.1  p =  3.9e-118   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(4,14-0)     R=+118.1  p =  6.5e-109   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(5,14-0)     R=+127.9  p =  6.2e-118   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(6,14-0)     R=+125.7  p =  5.6e-116   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(7,14-0)     R=+126.1  p =  2.5e-116   FAIL !!!!!
  [Low1/64]FPF-14+6/16:(8,14-1)     R= +92.5  p =  1.0e-81    FAIL !!!!
  [Low1/64]FPF-14+6/16:(9,14-2)     R= +66.4  p =  8.1e-58    FAIL !!!!
  [Low1/64]FPF-14+6/16:(10,14-2)    R= +68.8  p =  5.5e-60    FAIL !!!!
  [Low1/64]FPF-14+6/16:(11,14-3)    R= +40.8  p =  1.6e-35    FAIL !!!
  [Low1/64]FPF-14+6/16:(12,14-4)    R= +32.1  p =  3.5e-26    FAIL !!
  [Low1/64]FPF-14+6/16:(13,14-5)    R= +16.4  p =  1.8e-13    FAIL
  [Low1/64]FPF-14+6/16:(14,14-5)    R= +27.1  p =  2.6e-22    FAIL !!
  [Low1/64]FPF-14+6/16:(15,14-6)    R= +15.6  p =  5.6e-12   VERY SUSPICIOUS
  [Low1/64]FPF-14+6/16:(16,14-7)    R= +11.1  p =  1.1e-8   very suspicious
  [Low1/64]FPF-14+6/16:(17,14-8)    R=  +9.1  p =  1.3e-6   unusual
  [Low1/64]FPF-14+6/16:(18,14-8)    R= +10.6  p =  1.2e-7   suspicious
  [Low1/64]FPF-14+6/16:all          R=+388.5  p =  4.1e-364   FAIL !!!!!!!
  [Low1/64]FPF-14+6/16:all2         R=+30616  p = 0           FAIL !!!!!!!!
  [Low1/64]FPF-14+6/16:cross        R=  +5.7  p =  3.8e-5   mildly suspicious
  ...and 96 test result(s) without anomalies

(END)
```

Figure A.2: Practrand failure result of ChaCha in Wolfcrypt library.

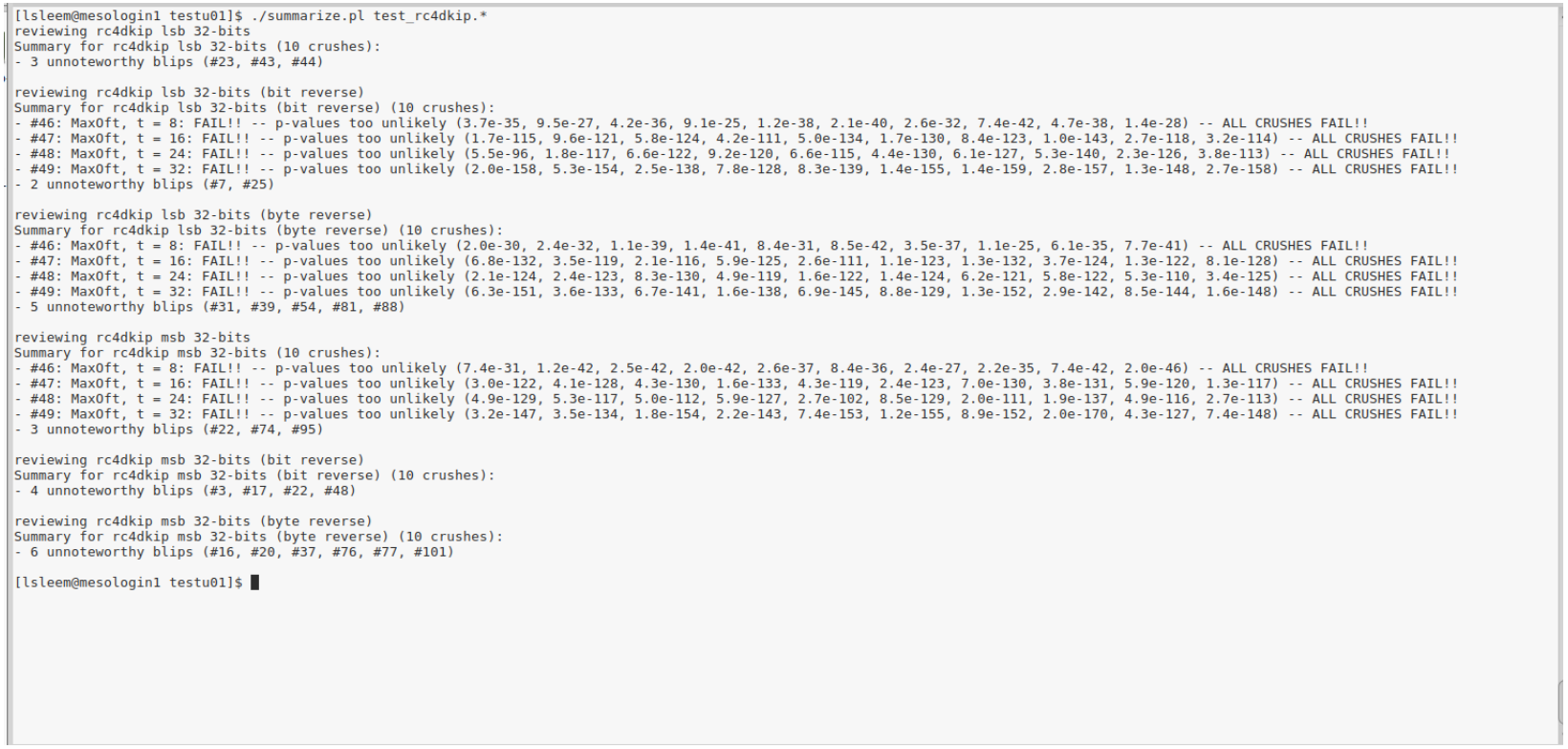
```
rng=RNG_stdin64, seed=0x180361a9
length= 8 gigabytes (2^33 bytes), time= 360 seconds
  Test Name                         Raw       Processed     Evaluation
  [Low16/64]BCFN(2+6,13-3,T)        R= +10.9  p =  5.7e-5   unusual
  ...and 211 test result(s) without anomalies

rng=RNG_stdin64, seed=0x180361a9
length= 16 gigabytes (2^34 bytes), time= 715 seconds
  no anomalies in 223 test result(s)

rng=RNG_stdin64, seed=0x180361a9
length= 32 gigabytes (2^35 bytes), time= 1419 seconds
  no anomalies in 233 test result(s)

rng=RNG_stdin64, seed=0x180361a9
length= 64 gigabytes (2^36 bytes), time= 2836 seconds
  no anomalies in 244 test result(s)

rng=RNG_stdin64, seed=0x180361a9
length= 128 gigabytes (2^37 bytes), time= 5655 seconds
  no anomalies in 255 test result(s)

rng=RNG_stdin64, seed=0x180361a9
length= 256 gigabytes (2^38 bytes), time= 11415 seconds
  no anomalies in 265 test result(s)

rng=RNG_stdin64, seed=0x180361a9
length= 512 gigabytes (2^39 bytes), time= 22738 seconds
  no anomalies in 276 test result(s)

rng=RNG_stdin64, seed=0x180361a9
length= 1 terabyte (2^40 bytes), time= 45289 seconds
  no anomalies in 287 test result(s)

rng=RNG_stdin64, seed=0x180361a9
length= 2 terabytes (2^41 bytes), time= 92613 seconds
  Test Name                         Raw       Processed     Evaluation
  BCFN(2+0,13-0,T)                  R= +10.3  p =  4.8e-5   mildly suspicious
  ...and 296 test result(s) without anomalies

rng=RNG_stdin64, seed=0x180361a9
length= 4 terabytes (2^42 bytes), time= 182597 seconds
  Test Name                         Raw       Processed     Evaluation
  BCFN(2+0,13-0,T)                  R= +24.3  p =  1.6e-12    FAIL
  ...and 307 test result(s) without anomalies

(END)
```

(a)

```
rng=RNG_stdin64, seed=0xaa6b302b
length= 8 gigabytes (2^33 bytes), time= 377 seconds
  Test Name                         Raw       Processed     Evaluation
  [Low16/64]BCFN(2+6,13-3,T)        R= +10.9  p =  5.7e-5   unusual
  ...and 211 test result(s) without anomalies

rng=RNG_stdin64, seed=0xaa6b302b
length= 16 gigabytes (2^34 bytes), time= 747 seconds
  no anomalies in 223 test result(s)

rng=RNG_stdin64, seed=0xaa6b302b
length= 32 gigabytes (2^35 bytes), time= 1483 seconds
  no anomalies in 233 test result(s)

rng=RNG_stdin64, seed=0xaa6b302b
length= 64 gigabytes (2^36 bytes), time= 2967 seconds
  no anomalies in 244 test result(s)

rng=RNG_stdin64, seed=0xaa6b302b
length= 128 gigabytes (2^37 bytes), time= 5924 seconds
  no anomalies in 255 test result(s)

rng=RNG_stdin64, seed=0xaa6b302b
length= 256 gigabytes (2^38 bytes), time= 11771 seconds
  no anomalies in 265 test result(s)

rng=RNG_stdin64, seed=0xaa6b302b
length= 512 gigabytes (2^39 bytes), time= 23599 seconds
  no anomalies in 276 test result(s)

rng=RNG_stdin64, seed=0xaa6b302b
length= 1 terabyte (2^40 bytes), time= 47280 seconds
  no anomalies in 287 test result(s)

rng=RNG_stdin64, seed=0xaa6b302b
length= 2 terabytes (2^41 bytes), time= 94165 seconds
  Test Name                         Raw       Processed     Evaluation
  BCFN(2+0,13-0,T)                  R= +10.3  p =  4.8e-5   mildly suspicious
  ...and 296 test result(s) without anomalies

rng=RNG_stdin64, seed=0xaa6b302b
length= 4 terabytes (2^42 bytes), time= 188941 seconds
  Test Name                         Raw       Processed     Evaluation
  BCFN(2+0,13-0,T)                  R= +24.3  p =  1.6e-12    FAIL
  ...and 307 test result(s) without anomalies

(END)
```

(b)

Figure A.3: (a) Practrand failure result of RC4Dkip-Optimized version and (b) RC4Dkip-Plain version.



```
[lsleem@mesologin1 testu01]$ ./summarize.pl test_rc4dkip.*
reviewing rc4dkip lsb 32-bits
Summary for rc4dkip lsb 32-bits (10 crushes):
- 3 unnoteworthy blips (#23, #43, #44)

reviewing rc4dkip lsb 32-bits (bit reverse)
Summary for rc4dkip lsb 32-bits (bit reverse) (10 crushes):
- #46: MaxOft, t = 8: FAIL!! -- p-values too unlikely (3.7e-35, 9.5e-27, 4.2e-36, 9.1e-25, 1.2e-38, 2.1e-40, 2.6e-32, 7.4e-42, 4.7e-38, 1.4e-28) -- ALL CRUSHES FAIL!!
- #47: MaxOft, t = 16: FAIL!! -- p-values too unlikely (1.7e-115, 9.6e-121, 5.8e-124, 4.2e-111, 5.0e-134, 1.7e-130, 8.4e-123, 1.0e-143, 2.7e-118, 3.2e-114) -- ALL CRUSHES FAIL!!
- #48: MaxOft, t = 24: FAIL!! -- p-values too unlikely (5.5e-96, 1.8e-117, 6.6e-122, 9.2e-120, 6.6e-115, 4.4e-130, 6.1e-127, 5.3e-140, 2.3e-126, 3.8e-113) -- ALL CRUSHES FAIL!!
- #49: MaxOft, t = 32: FAIL!! -- p-values too unlikely (2.0e-158, 5.3e-154, 2.5e-138, 7.8e-128, 8.3e-139, 1.4e-155, 1.4e-159, 2.8e-157, 1.3e-148, 2.7e-158) -- ALL CRUSHES FAIL!!
- 2 unnoteworthy blips (#7, #25)

reviewing rc4dkip lsb 32-bits (byte reverse)
Summary for rc4dkip lsb 32-bits (byte reverse) (10 crushes):
- #46: MaxOft, t = 8: FAIL!! -- p-values too unlikely (2.0e-30, 2.4e-32, 1.1e-39, 1.4e-41, 8.4e-31, 8.5e-42, 3.5e-37, 1.1e-25, 6.1e-35, 7.7e-41) -- ALL CRUSHES FAIL!!
- #47: MaxOft, t = 16: FAIL!! -- p-values too unlikely (6.8e-132, 3.5e-119, 2.1e-116, 5.9e-125, 2.6e-111, 1.1e-123, 1.3e-132, 3.7e-124, 1.3e-122, 8.1e-128) -- ALL CRUSHES FAIL!!
- #48: MaxOft, t = 24: FAIL!! -- p-values too unlikely (2.1e-124, 2.4e-123, 8.3e-130, 4.9e-119, 1.6e-122, 1.4e-124, 6.2e-121, 5.8e-122, 5.3e-110, 3.4e-125) -- ALL CRUSHES FAIL!!
- #49: MaxOft, t = 32: FAIL!! -- p-values too unlikely (6.3e-151, 3.6e-133, 6.7e-141, 1.6e-138, 6.9e-145, 8.8e-129, 1.3e-152, 2.9e-142, 8.5e-144, 1.6e-148) -- ALL CRUSHES FAIL!!
- 5 unnoteworthy blips (#31, #39, #54, #81, #88)

reviewing rc4dkip msb 32-bits
Summary for rc4dkip msb 32-bits (10 crushes):
- #46: MaxOft, t = 8: FAIL!! -- p-values too unlikely (7.4e-31, 1.2e-42, 2.5e-42, 2.0e-42, 2.6e-37, 8.4e-36, 2.4e-27, 2.2e-35, 7.4e-42, 2.0e-46) -- ALL CRUSHES FAIL!!
- #47: MaxOft, t = 16: FAIL!! -- p-values too unlikely (3.0e-122, 4.1e-128, 4.3e-130, 1.6e-133, 4.3e-119, 2.4e-123, 7.0e-130, 3.8e-131, 5.9e-120, 1.3e-117) -- ALL CRUSHES FAIL!!
- #48: MaxOft, t = 24: FAIL!! -- p-values too unlikely (4.9e-129, 5.3e-117, 5.0e-112, 5.9e-127, 2.7e-102, 8.5e-129, 2.0e-111, 1.9e-137, 4.9e-116, 2.7e-113) -- ALL CRUSHES FAIL!!
- #49: MaxOft, t = 32: FAIL!! -- p-values too unlikely (3.2e-147, 3.5e-134, 1.8e-154, 2.2e-143, 7.4e-153, 1.2e-155, 8.9e-152, 2.0e-170, 4.3e-127, 7.4e-148) -- ALL CRUSHES FAIL!!
- 3 unnoteworthy blips (#22, #74, #95)

reviewing rc4dkip msb 32-bits (bit reverse)
Summary for rc4dkip msb 32-bits (bit reverse) (10 crushes):
- 4 unnoteworthy blips (#3, #17, #22, #48)

reviewing rc4dkip msb 32-bits (byte reverse)
Summary for rc4dkip msb 32-bits (byte reverse) (10 crushes):
- 6 unnoteworthy blips (#16, #20, #37, #76, #77, #101)

[lsleem@mesologin1 testu01]$
```

Figure A.4: TestU01 summarized failure result of RC4Dkip-Optimized version

```
[lsleem@mesologin1 testu01]$ ./summarize.pl test_rc4dkip_plain.*
reviewing rc4dkipplain lsb 32-bits
Summary for rc4dkipplain lsb 32-bits (10 crushes):
- 1 unnoteworthy blips (#12)

reviewing rc4dkipplain lsb 32-bits (bit reverse)
Summary for rc4dkipplain lsb 32-bits (bit reverse) (10 crushes):
- #46: MaxOft AD, t = 8: FAIL!! -- p-values too unlikely (4.0e-32, 1.1e-36, 3.3e-28, 1.0e-38, 4.0e-23, 2.7e-28, 8.0e-27, 1.1e-33, 1.2e-36, 1.0e-32) -- ALL CRUSHES FAIL!!
- #47: MaxOft, t = 16: FAIL!! -- p-values too unlikely (2.5e-135, 2.2e-115, 9.9e-125, 3.4e-120, 8.2e-117, 4.2e-140, 6.9e-123, 8.3e-143, 6.9e-143, 2.5e-127) -- ALL CRUSHES FAIL!!
- #48: MaxOft, t = 24: FAIL!! -- p-values too unlikely (3.9e-124, 4.7e-121, 4.7e-135, 8.0e-126, 5.4e-122, 5.2e-122, 1.4e-113, 2.6e-140, 1.1e-131, 2.6e-121) -- ALL CRUSHES FAIL!!
- #49: MaxOft, t = 32: FAIL!! -- p-values too unlikely (1.2e-151, 3.1e-151, 4.3e-143, 1.6e-139, 3.1e-148, 1.0e-130, 1.8e-152, 6.4e-146, 1.6e-147, 3.5e-130) -- ALL CRUSHES FAIL!!
- 1 unnoteworthy blips (#81)

reviewing rc4dkipplain lsb 32-bits (byte reverse)
Summary for rc4dkipplain lsb 32-bits (byte reverse) (10 crushes):
- #46: MaxOft AD, t = 8: FAIL!! -- p-values too unlikely (1.8e-52, 3.9e-30, 2.4e-35, 1.2e-36, 3.8e-33, 1.6e-29, 1.0e-41, 1.7e-30, 1.9e-25, 3.3e-40) -- ALL CRUSHES FAIL!!
- #47: MaxOft, t = 16: FAIL!! -- p-values too unlikely (1.0e-133, 6.3e-118, 1.2e-123, 9.0e-115, 1.6e-121, 9.4e-134, 8.2e-126, 4.8e-117, 2.7e-136, 1.8e-122) -- ALL CRUSHES FAIL!!
- #48: MaxOft, t = 24: FAIL!! -- p-values too unlikely (5.1e-140, 8.4e-126, 3.8e-128, 2.0e-135, 2.7e-118, 5.3e-128, 2.0e-122, 1.3e-115, 1.6e-128, 1.8e-111) -- ALL CRUSHES FAIL!!
- #49: MaxOft, t = 32: FAIL!! -- p-values too unlikely (5.7e-152, 1.7e-165, 4.5e-159, 5.6e-134, 3.0e-154, 5.6e-173, 3.2e-149, 2.1e-115, 1.2e-143, 8.5e-135) -- ALL CRUSHES FAIL!!
- 2 unnoteworthy blips (#87, #88)

reviewing rc4dkipplain msb 32-bits
Summary for rc4dkipplain msb 32-bits (10 crushes):
- #46: MaxOft AD, t = 8: FAIL!! -- p-values too unlikely (8.6e-30, 1.3e-35, 1.4e-33, 2.5e-35, 8.1e-37, 1.0e-35, 4.0e-40, 1.4e-33, 3.7e-27, 4.6e-31) -- ALL CRUSHES FAIL!!
- #47: MaxOft, t = 16: FAIL!! -- p-values too unlikely (4.2e-112, 1.1e-127, 7.9e-132, 3.4e-119, 2.4e-133, 5.8e-130, 2.0e-118, 5.1e-114, 1.2e-118, 4.7e-111) -- ALL CRUSHES FAIL!!
- #48: MaxOft, t = 24: FAIL!! -- p-values too unlikely (1.0e-118, 5.8e-128, 2.3e-125, 1.5e-104, 2.6e-135, 1.8e-150, 1.3e-124, 2.2e-116, 4.6e-115, 7.6e-147) -- ALL CRUSHES FAIL!!
- #49: MaxOft, t = 32: FAIL!! -- p-values too unlikely (3.0e-166, 1.9e-139, 2.3e-161, 1.7e-134, 1.7e-154, 9.3e-140, 8.5e-149, 2.8e-139, 3.6e-156, 7.8e-143) -- ALL CRUSHES FAIL!!
- 3 unnoteworthy blips (#11, #23, #61)

reviewing rc4dkipplain msb 32-bits (bit reverse)
Summary for rc4dkipplain msb 32-bits (bit reverse) (10 crushes):
- 5 unnoteworthy blips (#23, #23, #32, #76, #86)

reviewing rc4dkipplain msb 32-bits (byte reverse)
Summary for rc4dkipplain msb 32-bits (byte reverse) (10 crushes):
- 3 unnoteworthy blips (#23, #49, #90)

[lsleem@mesologin1 testu01]$
```

Figure A.5: TestU01 summarized failure result of RC4Dkip-Plain version

B

ANNEXE B

```
#include "splitmix64.h"

#define size  8
typedef char byte;
byte  plain[size];
byte  cipher[size];
u32 i,nonce[2],K[3],key[26];
unsigned char *k[12];
uint64_t myseed;

void speck_seed(uint64_t newseed) {
    myseed = newseed;

  for (int i=0; i<12 ; i++){
    k[i]=splitmix64_stateless(myseed  + i);
    myseed+=3 ;
  }

  for (int i=0; i<2; i++){
    nonce[i]=splitmix64_stateless(myseed  + i);
    myseed+=2;
  }

  for (int i=0; i<size; i++)
    plain[i]=0;

  for(i=0;i<3;i++) K[i]=((u32 *)k)[i];
    ExpandKey(K,key);
 }

uint64_t speck(void) {

 static int need_generate=0;
 uint64_t *res=(uint64_t*) cipher;

  if(need_generate==0) {
    need_generate=size/8;

    u64 *newnonce=(u64*)nonce;
    (*newnonce)++;

    speck6496(cipher,plain,sizeof(plain),nonce, key);

 }
  need_generate--;
```

```
return res[size/8-1-need_generate];
}
```

Listing B.1: C Code for Speck Randomness Test

```
#include "splitmix64.h"

byte plain[size];
byte cipher[size];
uint64_t  myseed;
u32  Nonce[2], i,K[3],key[26],subkey[16];
unsigned char *k[12];

void Speck-R_seed(uint64_t newseed) { myseed = newseed;

  Sbox1=malloc(sizeof(uchar)*256);
  Sbox2=malloc(sizeof(uchar)*256);
  Sbox3=malloc(sizeof(uchar)*256);
    Sboxnew=malloc(sizeof(uchar)*256);
    DK=malloc(sizeof(uchar)*256);

  for (int i=0; i<2; i++){
    Nonce[i]=splitmix64_stateless(myseed + i);
    myseed+=2;
    }

  for (int i=0; i<12; i++){
    k[i]=splitmix64_stateless(myseed + i);
    myseed+=12;
    }

  for (int i=0; i<16; i++){
    subkey[i]=splitmix64_stateless(myseed + i);
    myseed+=16;
    }

  for (int i=0; i<size; i++)
   plain[i]=0;

  for (int i=0; i<256; i++){
   DK[i]=splitmix64_stateless(myseed + i);
     myseed+=256;
    }


 for(i=0;i<3;i++) K[i]=((u32 *)k)[i];

 ExpandKey(K,key);

 rc4key(&DK[0],Sbox1,64);
 rc4key(&DK[64],Sbox2,64);
 rc4key(&DK[128],Sbox3,64);

 }

uint64_t Speck-R(void) {

  static int need_generate=0;
  uint64_t *res=(uint64_t*) cipher;
  static iter=0;
  static iter2=0;
```

```
57
58   if(need_generate==0) {
59     need_generate=size/8;
60
61    u64 *newnonce=(u64*)Nonce;
62     (*newnonce)++;
63
64    Speck6496-R(cipher, plain, sizeof(plain), Nonce, key);
65
66     iter++;
67     iter2++;
68
69     if(iter==2000) {
70
71       for(int i=0;i<256;i++) {
72         Sbox1[i]=Sbox2[Sbox1[i]];
73       }
74
75       if(iter2==2000*2000) {
76         for(int i=0;i<256;i++) {
77           Sbox2[i]=Sbox3[Sbox2[i]];
78         }
79         iter2=0;
80       }
81
82
83       iter=0;
84     }
85
86   }
87   need_generate--;
88   return res[size/8-1-need_generate];
89 }
```

Listing B.2: C Code for Speck-R Randomness Test

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — http://www.multiagent.fr/ThesisStyle
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/