REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'Informatique

N° d'ordre:
N° Série:

Thèse de Doctorat en Sciences en Informatique

Thème

Problèmes de Sécurité dans les Systèmes Embarqués

(Security Problems in Embedded Systems)

Présentée par : Chikouche Noureddine

Date de soutenance: 17 mars 2016

Devant le jury composé de:

Djedi Noureddine	**Professeur, Université de Biskra**	**Président**
Cherif Foudil	**Professeur, Université de Biskra**	**Rapporteur**
Benmohammed Mohamed	**Professeur, Université de Constantine 2**	**Co-Rapporteur**
Bilami Azeddine	**Professeur, Université de Batna**	**Examinateur**
Chaoui Allaoua	**Professeur, Université de Constantine 2**	**Examinateur**
Babahenini Mohamed Chaouki	**Maître de Conférences A, Université de Biskra**	**Examinateur**

Acknowledgement

My deep gratitude goes first to *Allah* Who helped me to fulfil this research

I would like to thank my honourable advisor **Pr. Foudil Cherif** for his guidance, advice and encouragement. I also thank my honourable co-advisor **Pr. Mohamed Benmohammed** for his advice. I am grateful for the good times we had over the years. We had inspiring discussions, from which the ideas that solidified into this thesis emerged eventually.

I wish to thank the board of examiners **Pr. Djedi Noureddine**, **Pr. Bilami Azeddine**, **Pr. Chaoui Allaoua**, and **Dr. Babahenini Mohamed Chaouki** for accepting to evaluate this thesis.

Moreover, I'd like to thank **Dr. Pierre-Louis Cayrel**, Hubert-Curien Laboratory, Saint-Etienne, France for his advice, sharing his ideas, and collaborating with us, which lead to a significant part in this thesis. My thanks also go to the staff of *Applied Cryptography & Telecom* group, Hubert-Curien laboratory especially **Pr. Viktor Fischer** and **Mme Nathalie Bochar** for their cooperation with me during the stays I spent in France.

I wish to thank all my colleagues and the staff of the computer science department at both M'sila and Biskra Universities.

To everyone who enlightened me with a kind word, I extend my sincere appreciation.

Finally and most importantly, my biggest thanks go to my parents, my wife, my children, and my family, for their love and support during times of hardship.

الملخص

أنظمة التعرف بترددات الراديو هي واحدة من أهم الأنظمة المشحونة والتي عرفت تطورا سريعا في السنوات الأخيرة. هذه الأنظمة تستعمل في العديد من التطبيقات، مثل الصحة والنقل ومراقبة العبور إلخ. من ناحية أخرى، اعتماد هذه التكنولوجيا على ترددات الراديو في التواصل، يؤدي إلى مشاكل في الأمن والخصوصية.

على مستوى الأبحاث التي تهتم بتصميم وإنجاز بروتوكولات المصادقة الخاصة بأنظمة التعرف بترددات الراديو (أ.ت.ت.ر)، نجد عدد معتبر من البروتوكولات التي تستعمل مختلف الأشكال الجبرية والتشفيرية (التشفير المتماثل، التشفير غير المتماثل، دوال التقطيع، معاملات البيتات، إلخ). محدودية الموارد في الرقاقة منخفضة الكلفة ولا تسمح بتنفيذ كل خوارزميات تشفير. من بين أهم الأشكال التشفيرية المستعملة لتامين البروتوكولات، نذكر دوال التقطيع والتشفير المعتمد على الأكواد. هذا الأخير ليس سريعا فحسب، بل ويقاوم الهجمات الكوانتية ولا يحتاج إلى معالج خاص بالتشفير.

نقوم في أطروحتنا بتحليل عدد من البروتوكولات الحديثة. نقترح بروتوكول مصادقة أ.ت.ت.ر موجه لتطبيقات مراقبة العبور مستعملا دوال التقطيع التشفيرية و دوال التقطيع البيوميترية. بعد ذلك نقترح بروتوكولين محسنين يعتمدان على شكلين لخوارزم التشفير ماك إيليس، الشكل الأول هو خوارزم التشفير ماك إيليس العشوائي، والثاني هو خوارزم التشفير ماك إيليس المتجه نحو الحلقية مع تحقيق شبه كثافة معتدلة (QC-MDPC).

يحوي عملنا أيضا على مقارنة بين البروتوكولات التي إقترحناها مع مختلف البروتوكولات الموجودة والمعتمدة على أكواد تصحيح الخطأ من حيث الأمن والفعالية. نتحقق من خواص الأمن باستعمال تطبيقات أفيسبا (AVISPA) ونتحقق من خاصية الخصوصية باستعمال النموذج المقترح من وافي وفان. كذلك نقوم بتحليل فعالية البروتوكولات المقترحة من حيث المساحة اللازمة في الذاكرة وتكلفة الاتصال وتكلفة الحساب. البروتوكولات التي اقترحناها فعالة ولا تحتاج إلى البحث الشامل ويمكن للرقاقة تنفيذ العمليات التشفيرية قليلة التكلفة.

الكلمات المفتاحية: الأنظمة المشحونة، أنظمة التعرف بترددات الراديو، بروتوكولات المصادقة، التشفير المعتمد على الأكواد، الأمن.

Abstract

Radiofrequency identification (RFID) systems are among the most important embedded systems that saw fast evolutions during the last years. These systems are used in several applications, such as, health, transportation, access control, etc. However, the communication in this technology is based on radio waves, which poses problems in security and privacy.

In the literature of design and implementation of authentication protocols on RFID systems, we can find many protocols developed using various algebraic and cryptographic primitives (asymmetric cryptosystems, symmetric cryptosystems, hash function, bitwise operators, etc.). The limitation of resources (e.g. memory, computation, etc.) on low-cost RFID tags does not permit the implementation of all the cryptosystems. Among cryptographic primitives used to secure the RFID authentication protocols, we cite code-based cryptography. It is very fast, it resists quantum attacks, and does not require any crypto-processor.

In our thesis, we analyse the security of several recent RFID authentication protocols. We propose a new RFID protocol oriented towards access control applications. It uses cryptographic hash function and Biometric hash function. After that, we propose two improved protocols based on two variants of McEliece encryption scheme, the first is the randomized McEliece cryptosystem, and the second is Quasi Cyclic-Moderate Density Parity Check (QC-MDPC) McEliece cryptosystem.

Our work also includes a comparison between our proposed protocols and different existing protocols based on error-correcting codes in terms of security and performance. Security properties are proved by AVISPA (*Automated Verification Internet Protocol and its Applications*) tools, and the privacy property is verified by Ouafi-Phan model. The Performance of proposed protocols is analysed in terms of storage requirement, communication cost and computational cost. The performance of our protocols are effective, don't need to do exhaustive search, and the tag can perform lightweight cryptographic operations.

Keywords: Embedded systems, RFID, Authentication protocols, Code-based cryptography, Security

Résumé

Les systèmes d'identification par radiofréquence (RFID) sont des systèmes embarqués qui ont connu des développements rapides dans les dernières années. Ces systèmes sont utilisés dans plusieurs applications, telles que : santé, transport, contrôle d'accès, etc. Cependant, la communication dans cette technologie est basée sur les ondes radio, ce qui crée des problèmes de sécurité et de vie privée.

Dans les travaux de recherche qui s'intéressent à la conception et l'implémentation des protocoles d'authentification des systèmes RFID, on peut trouver plusieurs protocoles en utilisant différentes primitives cryptographiques et algébriques (telles que : cryptosystèmes symétriques, cryptosystèmes asymétriques, fonctions de hachage, opérateurs des bits, etc.). La limitation des ressources (e.g. mémoire, capacité de calcul, etc.) dans les tags bas coût ne permet pas d'implémenter tous les cryptosystèmes. Parmi les primitives utilisées pour sécuriser les protocoles d'authentification, on cite la cryptographie basée sur les codes. Elle est très rapide, résistante aux attaques quantiques, et n'exige pas de crypto-processeur.

Dans notre thèse, on analyse la sécurité de plusieurs protocoles d'authentification RFID récents. On propose un nouveau protocole RFID pour les applications de contrôle d'accès. Celui–ci utilise la fonction de hachage cryptographique et la fonction de hachage biométrique. Ensuite, on propose deux protocoles améliorés qui sont basés sur deux variantes de cryptosystème McEliece, la première est le cryptosystème McEliece aléatoire, et la deuxième est le cryptosystème McEliece basé sur QC-MDPC (*Quasi Cyclic-Moderate Density Parity Check*).

Notre travail consiste aussi à établir une comparaison entre nos deux protocoles et les différents protocoles existants basés sur les codes des correcteurs d'erreurs en termes de sécurité et de performance. Les propriétés de sécurité sont prouvées par les outils de AVISPA (*Automated Verification Internet Protocol and its Applications*), et la propriété de vie privée est vérifiée par le modèle de Ouafi-Phan. La performance des protocoles proposés est analysée en termes d'espace de stockage exigé et de coût de communication et de calcul. La performance de nos protocoles est effective, n'exige pas la recherche exhaustive, et le tag peut exécuter les opérations cryptographiques légères.

Mots-clés **:** Systèmes embarqués, RFID, protocoles d'authentification, cryptographie basée sur les codes, sécurité.

Contents

Chapter 2: RFID Systems and their Security

Chapter 3: Algebraic Replay Attacks

LIST OF FIGURES

LIST OF TABLES

NOTATIONS

R, T, S	Reader, Tag, and Server, respectively
$\mathcal{A}$	Adversary (or intruder)
RC	Registration center
$h(.)$	One-way hash function
$g(.)$	Pseudo-random function
$\|$	Concatenation of two inputs
id	The unique identifier of a tag
RID	The unique identifier of a reader
$\oplus$	Or-exclusif operator (XOR)
X_{old}	secret synchronization, old value of datum X
X_{new}	secret synchronization, new value of datum X
N_T	Random number generated by tag
N_{db}	Random number of server databases
N_R	Random number generated by reader
s,x,y	Secret value
$RH(.)$	Right-half of input message
$LH(.)$	Left-half of input message
$CRC(.)$	Cyclic Redundancy Check
B	Biometric template
$h_B(.)$	BioHash (Biometric hash function)
GB	Biohashed value of B
$\mathcal{C}(n,k,d)$	Binary linear code, where n is length and k is dimension which stands a generator matrix G' and d is the minimum distance

G	Public-key matrix in McEliece cryptosystem and its variants
G'	Generator matrix
H	Public-key matrix in Niederreiter cryptosystem and its variants
H'	Parity check matrix of binary code C
$\psi(.)$	a polynomial-time decoding algorithm
G_1, G_2	matrices with k_1xn and k_2xn, respectively,
t, t'	Two integer numbers
rand, *rand*'	Secret random numbers
$\phi(.)$	decoding application (transform x into error vector e)
e	Error vector of length n and weight $t'<t$ where $t=\lfloor(d-1)\rfloor/2$
$Right(e,k_1)$	Extract a substring from e, starting from the right-most bit, with length k_1
c_r, $c_{r'}$	Codewords, where $c_r=rG_1$ and $c_{r'}=r'G_1$
c_{id}	Codeword, where $c_{id}=idG_2$
DID	Dynamic ID, codeword with length n, where $DID=c_r \oplus c_{id}$

Introduction

In our modern life, we cannot find anything that does not use embedded devices. For example, we have a new type homes called smarthomes where all rooms, things (e.g. TV, light, mobile, etc.) and networks (e.g. gas, electricity, etc.) are connected and exploited by embedded systems (e.g. sensor, Wi-Fi, ultrared, Radio frequency identification, etc.) to carry out the services of the owner. The use of things is not limited to the home only, but one can execute any command by remote control. This is a new topic of research named Internet of things (IoT).

The development of embedded systems is articulated around two sides: the performance and the security. Concerning the performance, the main aims of designers of embedded systems are to: minimize the required memory (permanent and volatile), accelerate computation, optimize consummation of energy, and minimize communication cost between the entities of the system. The security is an important challenge in embedded systems and especially after the development of new cryptanalysis algorithms and the emergence of quantum computers. The study of security is depending on the system layers (application, communication, physic).

In this work, we cannot study all the embedded systems and all the mechanisms of security as it is a very vast research domain. We interest ourselves in an important embedded system which is used in IoT and applied in different applications (access control, health, shopping, transportation, etc.), that is Radio Frequency Identification (RFID). In security, we study an important area of research; it is design, verification and implementation of authentication protocols. This area is considered to be very critical area.

A typical RFID system consists of three components: the server, the reader, and the tag. The communication channel between the tag and the reader is based on radio frequency waves; it is unsecure, since it is open to attacks on authentication protocol. It is particularity the case of cryptographic protocol. In survey of RFID authentication protocols, there are an important number of authentication protocols which use different cryptographic primitives: private-key cryptosystems [FDW04, SOF05], hash functions [LAK06, Liu08, WHC11, JDTL12, Khe14], algebraic primitives [PCMA06, Chi07, Zen09], public-key cryptosystems (PKC) [MM12, Chien13, HKCL14, XPK14, LYL14, KGA15]. To design an authentication protocol, one chooses the required cryptographic and

algebraic primitives which are compatible with available resources of system's components, and one specifies security and privacy properties. Before implementing this protocol, it must be proved by formal tools.

Modern cryptosystems are divided into two classes, private-key cryptosystems and public-key cryptosystems. The first one is fast, but the major problem is the exchange key. In the second class, the problem of exchange key is not posed, because it uses a notion of pairs key: the public-key to encryption of plaintext and the private-key to decryption of ciphertext. The security of PKC is based on different building theory. We cite two categories of public-key cryptosystems, PKC based on number theory and PKC based on coding theory.

The PKC based on number theory uses a hard arithmetic problem, such as factorisation problem and discreet logarithm problem. The performance of this class of cryptosystems is not compatible with available resources of RFID systems. In addition, it does not resist quantum attacks; here we cite that the first commercial quantum computer will be available for everyone in 2020 [Eva09], it's crucial to improve the security protocols and cryptosystems which are used to protect the information in communication.

The second one is based on coding theory, is based on difficult problems NP-complete (syndrome decoding, etc.) and it resists quantum attacks. It does high-speed encryption and decryption compared to other public-key cryptosystems. It does not require a crypto-processor, and it uses different schemes, such as, public-key encryption scheme, identification schemes, secret sharing and signature. The major problem has been the size of public key. Recently, code-based cryptosystems were presented with small key sizes, for example, we quote [BCGO09, MB09].

The use of cryptographic primitives in low-cost RFID tags is limited because the space memory available is restricted, and the computational capabilities are limited. The lowest cost RFID tags are assumed to have the capability of performing bitwise operations (e.g. xor, and, etc.), bit shifts (e.g. rotate, logical shift, etc.) and random number generator (PRNG).

Contribution

In this thesis, we investigate the issues of security and privacy in low-cost RFID systems using hash function and code-based public key cryptography. The design of our

proposed protocols is based on avoiding the weaknesses of existing RFID protocols, validating the security and privacy requirements, and minimizing the required resources. All proposed protocols are verified by formal model and automated tools. The required resources in our protocols are compatible with available resources in low-cost tags. Our contributions in this thesis are:

- Describe in detail an important attack in RFID systems named Algebraic Replay Authentication Attack (ARAA). We analyse RFID authentication protocols where it does not resist to ARAA. We also propose a solution to avoid this attack.
- Propose a new protocol oriented to access control applications. This protocol is used in combined systems between RFID system and biometric system. It requires pseudo-random number generator (PRNG), biometric hash function and cryptographic hash function.
- Explain the disadvantage of the use hash function in RFID as it is need of exhaustive search in database of backend. To avoid this, we agree on the code-based cryptosystem. Then, we review different code-based RFID authentication protocols. Among these protocols, we discover weaknesses on two recent protocols.
- Propose two improved protocols based on two variants of McEliece encryption scheme, the first is based on the randomized McEliece cryptosystem and the second is based on Quasi Cyclic-Moderate Density Parity Check (QC-MDPC) McEliece cryptosystem.
- To verify security properties, all our proposed protocols are specified by HLPSL (High Level Language Specification Protocol) [Avi06] and proved by formal tools called AVISPA tools (Automated Verification Internet Protocol and its Applications) [ABBC+05].
- To prove the untraceability property, we use the privacy's model, which is proposed by Ouafi and Phan [OP08].

Thesis organization

This thesis contains a background and state-of-the-art study, a description of the proposed contributions, and some conclusions and perspectives.

This content is organized in 6 chapters as follows:

In chapter 1, we begin by describing the principal concepts of cryptography and specially concepts of public-key cryptography. We also show the important notions of coding theory and its applications. Finally, we present different code-based encryption schemes and critical attacks on McEliece cryptosystem.

In chapter 2, we show the RFID systems and their applications, we also describe different families of RFID systems and RFID authentication protocols. We portray security and privacy requirements, then explain different threats possible in these systems.

In chapter 3, we verify two RFID protocols by automated tools. The common characteristic between these protocols is that they do not resist Algebraic Replay Attacks on Authentication. We explain the main cause of this attack. Then, we describe how to avoid it.

The chapter 4 proposes a new RFID authentication protocol. It is based on the combination of two systems, RFID and Biometric. Then, we verify it in terms of validation of security and privacy properties. After that, we do a comparative study with other RFID protocols and biometric protocols.

In chapter 5, we show the different existing RFID authentication protocols based on errors-correcting codes. We prove the vulnerabilities of two recent RFID protocols. The first one is proposed by Malek and Miri [MM12] based on randomized McEliece cryptosystem. The second is proposed by Li et al. [LYL14] based on QC-MDPC (Quasi Cyclic-Moderate Density Parity Check) McEliece cryptosystem.

In the chapter 6, we propose improved versions of two studied protocols (Malek-Miri and Li et al.). It includes a comparison between the improved protocols and different protocols based on error-correcting codes in terms of security and performance. Security and privacy properties are prove, and the performance of the proposed improved protocols are analysed in terms of storage requirement, communicational cost and computational cost.

Finally, we end this thesis by a conclusion and perspectives, where we present our conclusive remarks and our suggestions for a future research.

Related publications

The results presented in this thesis were the subject of several publications in international journals, book chapters and in international conferences. Our scientific papers are listed hereafter in reverse chronological order.

- N. Chikouche, F. Cherif, P-.L. Cayrel, M. Benmohammed "A Secure Code-Based Authentication Scheme for RFID Systems,**"** *I.J. Computer Network and Information Security*, vol. 7, no. 9, pp. 1-9, 2015.
- N. Chikouche, F. Cherif, P-.L. Cayrel, M. Benmohammed "Improved RFID Authentication Protocol Based on Randomized McEliece Cryptosystem,**"** *International Journal of Network Security*, vol. 17, no. 4, pp. 413–422, 2015.
- N. Chikouche, F. Cherif, P-.L. Cayrel, M. Benmohammed "Weaknesses in Two RFID Authentication Protocols," *Codes, Cryptography and Information Security C2SI 2015,* S. El Hajji et al. (Eds.), LNCS, vol. 9084, pp. 162–172, Springer, 2015.
- N. Chikouche, F. Cherif, M. Benmohammed "Algebraic Replay Attacks on Authentication in RFID Protocols," *Advances in Security of Information and Communication Networks SecNet 2013*, A.I. Awad et al. (Eds.), CCIS, vol. 381, pp. 153–163, Springer, 2013.
- N. Chikouche, F. Cherif, M. Benmohammed, "Vulnerabilities of two Recently RFID Authentication Protocols," *The IEEE International Conference on Complex Systems (ICCS'12)*, IEEE, Agadir, Morocco, 2012.
- N. Chikouche, F. Cherif, M. Benmohammed, "An Authentication Protocol Based on Combined RFID-Biometric System", *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 3, No.4, pp. 62-67, 2012.
- N. Chikouche, F. Cherif, M. Benmohammed, "Conception et Vérification d'un Protocole d'Authentification de Système Combiné RFID-Biométrique », *CEUR Workshop Proceedings, Conférence Internationale sur l'Informatique et ses Applications (CIIA'11),* Saida, 2011.

Chapter 1

Cryptography and Coding Theory

1.1 Introduction

The main aim of cryptography is realizing the security properties (e.g. secrecy, authentication, etc.) by agreeing cryptographic primitives in messages transmitted between persons, organizations or states via computing devices (PC, server, mobile, etc.). The use of cryptography is not limited to diplomatic or military domains as in past; it has become important in different applications in modern life.

Among important cryptographic primitives are encryption schemes which comprise two main categories, private-key encryption schemes and public-key encryption schemes. The private-key encryption schemes date back from Caesar cryptosystem to AES (Advanced Encryption Scheme) cryptosystem. Concerning the second one, its first cryptosystem was proposed is RSA [RSA78]. The public-key encryption scheme is based on the hardness of number theoretic problems. However, P. Shor [Sho94] discovered that the quantum computers could solve the number theoretic problems, like factorization and discrete logarithm problems.

In this chapter, we will present the fundamental concepts and primitives of cryptography, and show different schemes of public-key cryptography, encryption scheme and signature scheme. We will concentrate on the most important ones which is the coding theory and its application in cryptography. We will show the principle concepts of coding theory, and we will discuss in detail the code-based encryption schemes which are McEliece cryptosystem and its variants and Niederreiter cryptosystems and its variants.

1.2 Cryptography

1.2.1 Private-key cryptography

The private key cryptosystem (symmetric-key cryptosystem) is a very old cryptosystem, it is used since antique. Its principle is: the encryption of a plaintext and the decryption of a ciphertext using the same key that is shared between two communicating entities (e.g. client, server). Before sending the ciphertext, it requires exchanging the private key by a predefined algorithm. The recent private-key cryptosystems is fast, doesn't require important space memory, it is implemented on hardware.

The major disadvantage of this category of cryptosystems is that the key must remain secret for all persons another one must legitimate entities. Then, it requires another algorithm to guarantee the exchange of the new key.

1.2.2 Public-Key cryptography

In public-key cryptography (PKC), the key of encryption and the key of decryption are different. Every entity possesses two distinct keys (private-key, public-key). The knowledge of the public key doesn't permit some to deduce the private key. Besides, it is impossible to deduce the key deprived from the public key. The public-key cryptography (or asymmetric cryptography) is based on a complex problem, i.e. difficult to resolve the problem. We found three families of problems, which are based on the hardness of lattice problems, which are based on number theory, and which are based on coding theory (see Figure 1.1). In this chapter, we interest by the two last categories.

In public-key cryptography based on number theory, the pair key is mathematically related. For example, the RSA cryptosystem [RSA78], which is proposed by Rivest, Shamir, and Adlmen in 1978, is based on the difficulty of factorization of two big numbers. Let p and q be two big prime numbers (e.g. with lengths 2048 bits), we can compute $n=pq$, but the problem is: if we know n we cannot find the value of p and q. Other example, the problem of discrete logarithm which is used in Diffie Hellman Exchange key protocol [DH76] and in Elgamal cryptosystem [ElG85], where the computation of $x^a \ mod \ n$ is simple, but it is extremely difficult in practice to recover the good x number.

Among the disadvantages of this family, the computation of encryption/decryption is hard and doesn't resist the quantum computing. In 1994, P. Shor [Sho94] found quantum algorithms for factoring and discrete logarithm, and these can be used to break the widely

used RSA cryptosystem and Diffie-Hellman key-exchange protocol using a quantum computer.

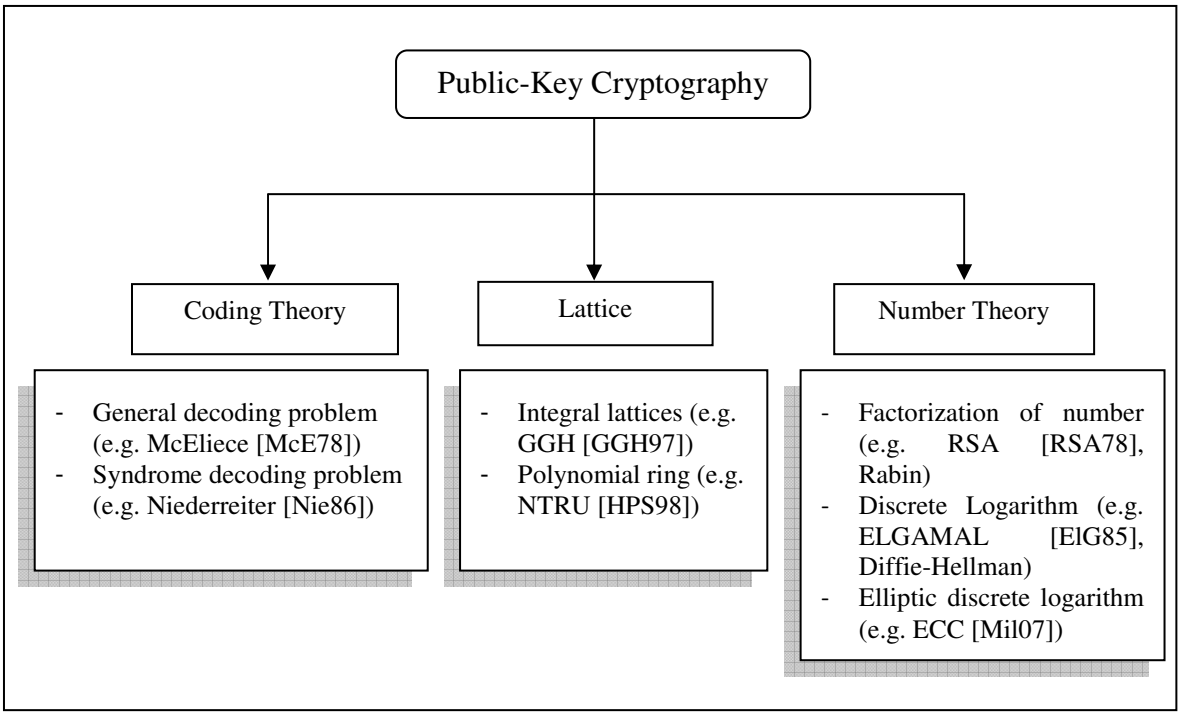

Figure 1.1: Categories of PKC

The public-key cryptosystem based on coding theory will be described in detail in section 1.4.

1.2.2.1 Public-key encryption scheme

This scheme permits to assure the confidential transmission of messages. If *Alice* wants to send encrypted message to *Bob*, she uses the public-key of *Bob* to encrypt the plaintext. In the other side, *Bob* uses his private-key to decrypt the received ciphertext. *Bob* is the only entity that can decrypt the ciphertext because he is only one to know the private-key.

Definition 1.1 (Public-key encryption scheme)

A *public-key encryption scheme is a triple*, (*G, E, D*), *of probabilistic polynomial-time algorithms which is as follows:*

Key generation algorithm (Gen) *a probabilistic expected polynomial-time algorithm G, which, on input the security parameter* 1^k *calculates a pair of keys (SK; PK) where SK is called the private key, and PK is the associated public key.*

An Encryption Algorithm (Enc) *is a deterministic polynomial time algorithm E which takes as input security parameter 1^k, a public-key PK and the plaintext m, and computes as output string c called the ciphertext. We use the notation $c= Enc_{PK}(m)$.*

Decryption Algorithm (Dec) *is a probabilistic polynomial time algorithm D that takes as inputs security parameter 1^k, a private-key SK, and a ciphertext c from the range of $E_{PK}(m)$, and computes as output a string m. We use the notation $m= Dec_{SK}(c)$.*

All algorithms should satisfy the completeness property, where for any pair of secret and public keys generated by key generation algorithm and any message m it holds that $D_{SK}(E_{PK}(m))=m$.

1.2.2.2 Public-key Signature scheme

The digital signature (sometimes named electronic) is a mechanism permitting to guarantee the integrity of a document (document cannot be modified but by the authorized entity) and to authenticate the author, and also the no-repudiation to the origin (to insure that a signatory won't be denied to have signature affixation to his document). Then, the digital signature has for goal to assure by computer tools means the same guarantees that a handwritten signature can provide.

The sender signs the document or the message by his private-key. This key is used to achieve the authentication of the sender and the integrity. The verification of validation of the documents is made by the public-key of the signatory. This key is used to achieve the non-repudiation property.

Definition 1.2 (Public-key signature scheme)

A *public-key signature scheme is a triple*, (*G, S, V*), *of probabilistic polynomial-time algorithms which is as follows:*

Key generation algorithm *a probabilistic expected polynomial-time algorithm G, which, on input security parameter 1^k calculates a pair of keys (SK; PK) where SK is the private key of signature generation, and PK is the associated public key of signature verification.*

Signature Algorithm *is a probabilistic polynomial time algorithm S which takes as input security parameter 1^k, a private-key SK of signatory and the message m. It*

returns the signature s of message m with private-key SK. We use the notation $s = S_{SK}(m)$.

Verification Algorithm *is a probabilistic polynomial time algorithm V that takes as inputs security parameter* 1^k*, a public-key PK of signatory, message m', and a signature s. It returns valid if s' is valid signature of message m' with the private-key corresponding PK and invalid else. We use the notation* $\{valid, invalid\} \leftarrow V_{PK}(s)$.

1.2.3 Security model

One of the most important objectives of an adversary is to obtain a simple information bit in plaintext correspondence of a given ciphertext. The notion correspondence is called *semantic security* or *indistinguishability* [GM82], and is symbolised by *IND*.

One considers that a cryptosystem is secure in terms of *indistinguishability*, in case of no adversary $\mathcal{A}$, given an encryption of a message randomly chosen from a two-element message space determined by the adversary, can identify the message choice with probability significantly better than that of random guessing (1/2). Therefore, this adversary is considered to have an *advantage* in distinguishing the ciphertext, if any intruder can succeed in distinguishing the chosen ciphertext with a probability significantly greater than 1/2.

Definition 2.3 (**IND-CPA**)

We say a public-key encryption scheme is ciphertext indistinguishable under chosen plaintext attacks (IND-CPA), if for every probabilistic polynomial time PPT-adversary $\mathcal{A}$ *has success-probability at most negligibly better than 1/2 in the experiment IND-CPA, i.e.* $Pr[IND\text{-}CPA(\mathcal{A}) = 1] \leq \frac{1}{2} + neg(\lambda)$.

IND-CPA security is modeled as the following game between the adversary and an experiment.

- The experiment generates a key pair, public and private keys (*PK, SK*).
- The public key *PK* is given to the adversary $\mathcal{A}$.
- The adversary chooses two plaintexts m_0 and m_1 of some length and provides them to the experiment.
- The experiment selects randomly a bit $b \leftarrow_{\$} \{0, 1\}$ and encrypts m_b. This ciphertext c^* is given to the adversary.
- The adversary has to guess whether the ciphertext contains m_0 or m_1.

- The adversary returns its estimation b'$\in\{0,1\}$. $\mathcal{A}$ wins if it guesses correctly (b=b').

One can summarize this game as follows:

***Experiment** IND-CPA*

(PK,*SK*) ← Gen(1^{λ})

(m_0,m_1, *state*) ← $\mathcal{A}_1(PK)$

b ← {0, 1}

c* ← $Enc_{PK}(m_b)$

b' ← $\mathcal{A}_2(c^*, state)$

if b=b' return 1 else return 0.

For example, the RSA cryptosystem in not semantically secure, and the ElGamal cryptosystem is semantically secure.

Naor and Yung [NY90] defined indistinguishability under (non-adaptive) chosen ciphertext attack (IND-CCA1) to model the capabilities of such stronger adversaries. The adversary is given access to a *decryption oracle* which decrypts arbitrary ciphertexts at the adversary's request, returning the plaintext.

Rackoff and Simon [RS91] proposed the notion of adaptive chosen (IND-CCA2). In the adaptive definition, the adversary gets access to a decryption oracle even after it has received a challenge ciphertext, with the restriction that it cannot use it to decrypt the challenge ciphertext. The last definition is the strongest of these three definitions of security.

For example, ElGamal cryptosystem is not CCA2 secure and RSA-OAEP (RSA with padding) is CCA2 secure in random oracle model.

1.2.4 Hash Function

Among the primitives used for data integrity and used in digital signature scheme, we cite the hash function or "one-way hash function".

Definition 1.4 (one-way function)

We say a function is one-way if it is easy to compute f(x) from x, but it is difficult to find x from y such as y=f(x).

A hash function takes like entry a non limited length value and sends back a value of *n* length fixed "hash value". For example, the length of SHA-1 is 160 bits and the length of MD-5 is 128 bits. The probability that a randomly chosen string gets mapped to a particular n-bit hash-value is 2^{-n}.

Hash functions must achieve three properties:

- *First pre-image resistance* A hash function is *first pre-image resistant* if, given a hash value *y*, where $h(x) = y$, it is hard to find any message *x*.
- *Second pre-image resistance* A hash function is *second pre-image resistant* if given a message *x*, it is hard to calculate a different value *x'* such that $h(x) = h(x')$. Sometimes called also weak collision-resistance.
- *Strong collision-resistance* A hash function is strong *collision resistant*, it is hard to find distinct inputs *x* and *x'* such that $h(x) = h(x')$.

1.2.5 Pseudo-Random Number Generator

A pseudo-random number generator (PRNG) is an algorithm that generates a sequence of numbers presenting some properties of the luck. For example, the numbers are supposed to be sufficiently independent from the some of the others, and it is potentially difficult to mark groups of numbers that follow a certain rule (behaviors of group).

Some pseudo-random number generators can be qualified as cryptographic when they show evidence of some necessary properties so that they can be used in cryptology. They must be capable of producing an exit sufficiently little discernible of an alea perfects and must resist attacks; for example the injection of forged data in order to produce some imperfections in the algorithm, or of the statistical analyses that would permit to predict the continuation.

1.2.6 Cryptographic protocol

The cryptographic protocol (or security protocol) is a set of exchange messages between the participants of a network, based on the cryptosystem notions that permit to secure the communications in a hostile environment by achieving certain security functionalities (secrecy, authentication, etc.).

We present the classes of protocols used with a limited number of participants and which assure specific goals.

Authentication protocol is a cryptographic protocol that assures the property of authenticity. The authentication is either unidirectional or mutual. We mention some protocols used extensively in the network communications: PGP (Pretty Good Privacy), Kerberos, and EAP (Expandable Authentication Protocol).

Exchange key protocol assures the generated symmetrical key confidentiality, shared by several participants, such as: the IKE protocol (Internet Key Exchange) and TLS (Transportation Layer Security).

Signature protocol The signatures of contract on Internet bring about two problems of security, no-repudiation and the fairness (i.e. to guarantee that no participant is penalized at the time of the signature of the contract). The objective of this protocol is to get to the signature of the contract distributed to an abuse free passage. An example of contract signature protocol is GJM [GJM99].

Zero-knowledge proof protocol The protocols of this class are destined to the proof of data indeed without revealing them. The first approach of zero-knowledge proof has been developed by A. Fiat and A. Shamir [FS86] in 1986. In the systems" zero-knowledge proof" the verifier does not need a secret and the prover possesses a varied secret that doesn't put in peril the whole system. It is a very powerful method to authenticate the messages, without giving the least information on the used secret, because a part is left at random.

1.3 Coding Theory

In domain of communication, if we send a message via a transmission channel (e.g. telephone, satellite, ADSL, etc.), the received message is not always the same as the emitted one, it exists an error rate. The error rate is the probability that a bit transmitted by the channel is different from the emitted bit. This error rate is different from a transmission channel to another. In the network computer, the error rate depends on the number of repetors and the type of channel (cable, ADSL, Wi-Fi, optic fibre, etc.). For communication with optic fiber, it can attain 10^{-9} (until one error for 10^{9} bits is transferred). The error rate is not only for the support of communication but also for the support of storage. In case of engrave file on CD or DVD, there exist errors in the file which is stored on CD/DVD. To resolve this problem, in 1950, Richard Hamming [Ham50] developed the premises of the codes theory.

The errors-correcting codes are a tool aiming to improve the reliability of the transmissions on a noisily channel. The method that they use consists in sending on the channel more data than the quantity of information to transmit. A redundancy is introduced thus. If this redundancy is structured in an exploitable manner, it is then possible to correct possible errors introduced by the channel. One can then, in spite of the noise, recover the entirety of the information transmitted at the departure.

We can find an important number of classes of error-correcting codes, but the most important class studied in literature are linear error correcting codes. In our work, we are interested in this class.

1.3.1 Linear Error Correcting Codes

Linearity allows efficient representation of codes and facilitates the analysis of their properties. Linear codes are subspaces of finite vector spaces. We study the finite field of which size is 2 symbolized by ($\mathbb{F}_2$). Then, our study is articulated on the binary linear code.

In this subsection, we present some notions on coding theory in order to clarify this topic. For more details, the reader is redirected to [MM77, Cay08, Hal10, RC14].

Definition 1.5 (Hamming weight)

The (Hamming) weight of a vector v is the number of non-zero entries. We use $\omega t(v)$ *to represent the Hamming weight of v.*

Definition 1.6 (Hamming distance)

The Hamming distance d(x, y) between the bit strings $x = x_1x_2\ldots x_n$ *and* $y = y_1y_{2\ldots}y_n$ *is the number of positions in which these strings differ, that is, the number of i (i = 1, 2,…, n) for which* $x_i \neq y_i$.

Definition 1.7 (Linear code)

A linear binary code of length n, dimension k and minimum distance d is denoted by $\mathcal{C}(n,k,d)$, *where k and n are positive integers with k<n.* $\mathcal{C}$ *is a t-error correcting linear code, that means the error-correcting capability of such a code is the maximum number t of errors that the code is able to decode.*

If ωt (.) denotes the Hamming weight for a linear code $\mathcal{C}$, then the Hamming distance *Dist*(.) is defined by the following formula:

$$\forall x, y \in \mathcal{C} \text{ Dist}(x, y) = \omega t\,(x - y)$$

The next increase is verified for all linear codes. It is called terminal Singleton:

$$n - k \geq d - 1$$

If the bound of Singleton is reached, the code is said MDS.

The minimum distance of a code tells us how many errors it can detect and how many errors it can correct, as the following two theorems show:

- A binary code $\mathcal{C}$ can detect up to k errors in any codeword if and only if $Dist(\mathcal{C}) \geq k + 1$.
- A binary code $\mathcal{C}$ can correct up to k errors in any codeword if and only if $Dist(\mathcal{C}) \geq 2k + 1$.

Definition 1.8 (Generator Matrix)

Let $\mathcal{C}$ be a linear code over $\mathbb{F}_2$. A generator matrix $G \in \mathbb{F}_2^{k\times n}$ of C is a matrix whose rows form a basis of C:

$$C = \{xG : x \in \mathbb{F}_2^k\}.$$

If $G = (I^k | A^{k\times(n-k)})$ where I^k is identity matrix with dimension k and $A^{k\times(n-k)}$ is matrix, then this matrix is systematic.

Let $\mathcal{C}(n, k, d)$ be a linear binary code and a binary string m with length k, which can be encoded to a codeword of n bits $c=mG$, where G is the generator matrix. The generator matrix G with k dimension and n length can generate all vectors in the code by taking all possible linear combinations of the rows of the generator matrix. An error vector e of length n and Hamming weight of $wt(e)$ is less than or equal to $\lfloor(d-1)/2\rfloor$ added to the codeword c results in a vector $c'=c\oplus e$.

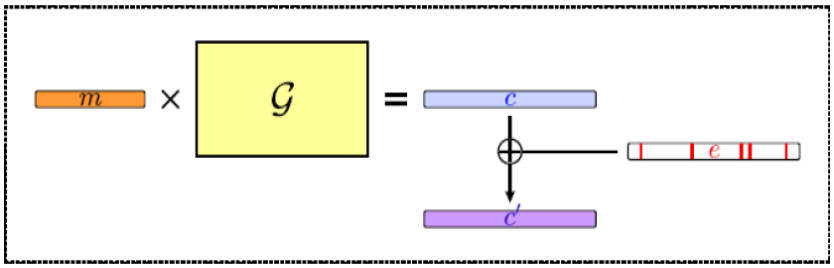

Figure 1.2: Encoding of message [Cay08]

Definition 1.9 (Equivalent codes)

We say that two (n, k) codes $\mathcal{C}$ and $\mathcal{C}'$ are equivalent, if there exists a permutation matrix $\boldsymbol{P} \in \mathbb{F}_2^{n\times n}$ such that $\mathcal{C}' = \boldsymbol{P}\,\mathcal{C}$.

Let G be a generator matrix of a (n, k) code $\mathcal{C}$ and G' is a generator matrix of a (n, k) code $\mathcal{C}'$. If two codes $\mathcal{C}$ and $\mathcal{C}'$ are equivalent, then there exists an invertible matrix T and a permutation matrix P such that:

$$\boldsymbol{G = PG'T}$$

Definition 1.10 (Dual code)

The dual code of $\mathcal{C}$, denoted $\mathcal{C}^{\perp}$. It is defined via scalar product:

$$\mathcal{C}^{\perp} = \{y \in \mathbb{F}_q^n | x.y = 0, \forall x \in \mathcal{C}\}$$

Definition 1.11 (Parity Check Matrix)

A parity check matrix H of $\mathcal{C}$ is an $(n-k) \times n$ matrix whose rows form a basis of the orthogonal complement of the vector subspace $\mathcal{C}$, i.e. it holds that,

$$\mathcal{C} = \{x \in F_2^n : H^t x = 0\}.$$

In general, suppose that G is a $k \times n$ generator matrix with $G = (I|A)$. To G we associate the parity check matrix H, where $H = (A^t|I^{n-k})$.

$\mathcal{C}$ is the core of H. $c \in \mathcal{C}$ if only if $H^t c = 0$.

$S = H^t c' = H^t c \oplus H^t e$ is the syndrome of error.

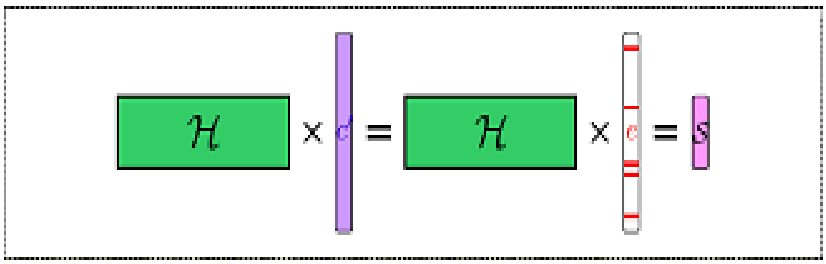

Figure 1.3: Syndrome of error [Cay08]

Decode consists in retrieving c from c'. Decoding algorithm γ_G is application:

$$\gamma_G: \quad \mathbb{F}_q^n \rightarrow \mathbb{F}_q^k$$

$$m \times G \oplus e \mapsto \begin{cases} m & if\ \ wt(e) \leq t \\ ? & if\ wt(e) > t \end{cases}$$

Decoding algorithm corrects t errors, if only if $\forall e \in \mathbb{F}_q^n \; \forall m \in \mathbb{F}_q^k$:

$$wt(e) \leq t \Rightarrow \gamma_G (m \times G \oplus e) = m.$$

There are numerous decoding methods: exhaustive research of error vector, exhaustive research of codeword, and syndrome method. The two first methods are exponential problems. The syndrome method is NP-complete; the application γ_H is defined as follows:

$$\gamma_H: \quad \mathbb{F}_q^{n-k} \rightarrow \mathbb{F}_q^n$$

$$H^t e \mapsto \begin{cases} e & if \; \exists e \in \mathbb{F}_q^n \mid wt(e) \leq t \\ ? & if \; \nexists \end{cases}$$

The decoding algorithm by syndrome is capable to correct t errors, if only if $\forall e \in \mathbb{F}_q^n$:

$$wt(e) \leq t \Rightarrow \gamma_H (H^t e) = e.$$

1.3.2 Structures and Codes

1.3.2.1 *Hamming Codes*

We define the Hamming code using parity check matrices.

Definition 1.12 (Hamming code)

A Hamming code of order r is a code generated when we take as parity check matrix H an $r \times (2r - 1)$ matrix with columns that are all the $2r - 1$ nonzero bit strings of length r in any order such that the last r columns form the identity matrix.

A Hamming code of order r contains 2^{n-r} codewords where $n = 2^r - 1$ and is a perfect code. The minimum distance of a Hamming code of order r is 3 whenever r is a positive integer.

1.3.2.2 *Cyclic Codes*

Definition 1.13 (Cyclic code)

An (n,k,d) linear code $\mathcal{C}$ is cyclic if whenever $(c_0, c_1, \ldots, c_{n-1})$ is a codeword in $\mathcal{C}$, then $(c_{n-1}, c_0, \ldots, c_{n-2})$ is also a codeword in $\mathcal{C}$.

It is convenient to convert codeword vectors $c = (c_0, c_1, \ldots, c_{n-1})$ of length n into *code polynomials* $c(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}$ of degree less than n. Note that the left-most bit in a codeword is associated with the constant term in the code polynomial. The shifted codeword $c'(x)$ has associated code polynomial:

$$c'(x) = c_{n-1} + c_0 x + c_1 x^2 + \ldots + c_{n-2} x^{n-1}$$

So c'(x) has degree less than n and is equal to the remainder when $xc(x)$ is divided by $x^n - 1$. We can define c'(x) by:

$$c'(x) = xc(x) \pmod{x^n - 1}$$

That is, *c'(x)* and *xc(x)* are equal in the ring of polynomials $F[x] \pmod{x^n-1}$,

1.3.2.3 *Goppa Code*

The Goppa code has been introduced by V.D. Goppa in 1970 [Gop70]. Goppa code may be used in the key generation of McEliece cryptosystem (see 1.4.1). Goppa code $\Gamma(g, \mathcal{L})$ is defined by the irreducible polynomial g of degree t over the finite field $\mathbb{F}_{2^m}$ and his support $\mathcal{L} = \{\alpha_0, \dots, \alpha_{n-1}\}$ of n elements which are not root of g. The parity matrix of $\Gamma(g, \mathcal{L})$ is obtained from the following matrix:

$$H = \begin{pmatrix} \frac{1}{g(\alpha_0)} & \cdots & \frac{1}{g(\alpha_{n-1})} \\ \vdots & & \vdots \\ \frac{\alpha_0^{t-1}}{g(\alpha_0)} & \cdots & \frac{\alpha_{n-1}^{t-1}}{g(\alpha_{n-1})} \end{pmatrix}$$

Each element of this matrix is then decomposed by m elements, placed in columns, using the projection of $\mathbb{F}_{2^m}$ in $\mathbb{F}_2^m$. One passes thus from matrix of size $t \times n$ to new parity matrix H of size $mt \times n$ over $\mathbb{F}_2$. Elements of code $\Gamma(g, \mathcal{L})$ will be therefore all elements c such as:

$$H \times c^T = 0$$

All square sub-matrix $t \times t$ of H is inversible because it is written as the multiplication of Vandermonde matrix and diagonal inversible matrix:

$$H = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ \alpha_0^{t-1} & \cdots & \alpha_{n-1}^{t-1} \end{pmatrix} \times \begin{pmatrix} \frac{1}{g(\alpha_0)} & & 0 \\ & \vdots & \\ 0 & & \frac{1}{g(\alpha_{n-1})} \end{pmatrix}$$

Therefore, for each polynomial g, there exists a binary Goppa code of length m to the number of field elements $n = 2^m$. The dimension of this code is equal to the number of field

elements minus the degree of the irreducible Goppa polynomial multiplied by the degree of irreducible polynomial used to create the finite field $k \geq n - mt$ capable of correcting any pattern of t or fewer errors. The minimal distance is at least equal $t+1$.

1.3.3 Difficult problems in coding theory

We present a list of some difficult problems related to the theory of the error-correcting codes. The following problems are not all the problems, as there exists several other problems which are found in practice.

Berlekamp, McEliece and van Tilborg showed in 1978 [BMT78] that the problem of the research of words of weight and fixed syndrome was a problem NP-complete. It is made out of the resolution of the system:

$$Hx=i,\ |x|=w$$

where H is a binary matrix, i is a given vector (syndrome of x) and w is a fixed integer (weight of x), x being the unknown.

Definition 1.14 **(Syndrome decoding problem (SD))**

Input: *Let H is a binary matrix (n-k,n), w is an integer, and* $s \in \mathbb{F}_2^{n-k}$ *is a syndrome*

Output: *word* $e \in \mathbb{F}_2^n$ $such\ that\ wt(e) \leq w\ and\ He^T = s$

This problem is used by Stern in his protocol, but some years later come out a variation of this problem called minimum distance (MD) which is NP-complete.

Definition 1.15 **(Minimum Distance problem (MD))**

Input: *Let H be a binary matrix (n-k,n), w is an integer* >0.

Question: *Does there exist a vector* $x \in \mathbb{F}_2^n$ *not null of weight* $\leq w$ *such that* $Hx^T=0$?

The Goppa Parameterized Bounded Decoding problem (GPBD) is a particular case of SD problem. This problem is also NP-complete.

Definition 1.16 **(Goppa Parameterized Bounded Decoding problem (GPBD))**

Input: *Let H be a binary matrix (n-k,n) (the parity matrix of Goppa code (n,k)) and a syndrome* $s \in \mathbb{F}_2^{n-k}$

Output: *word* $e \in \mathbb{F}_2^n$ $such\ that\ wt(e) \leq \frac{n-k}{\log_2 n}\ and\ He^T = s$

We cite another problem used in code-based cryptography which is Goppa Code Distinguishing problem (GD). This problem has been stated N. Courtois et al. in [CFS01].

Definition 2.17 (Goppa Code Distinguishing problem (GD)).

Input: *Let H be a binary matrix (n-k,n) (the parity matrix of Goppa code (n,k)) or random binary matrix (n-k,n)*

Output: b=1 *if* $H \in Goppa(n,k)$, b=0 *else*

T. Berger et al. [BCGO09] have proposed another decisional problem called Decoding by Quasi-cyclic syndrome. They proved that this problem is NP-complete.

Definition 2.18 (Decoding by Quasi-cyclic syndrome).

Being given $\ell > 1$*(one avoids the case* $\ell = 1$ *which correspond in degenerated case)* $A_1,\ldots, A_\ell$ *of size* $r^* \times n^*$ *over* $\mathbb{F}_q$*, an integer* $w < \ell n^*$ *and word* $z \in \mathbb{F}_q^{\ell r^*}$*. Let* $\ell r^* \times \ell n^*$ *matrix defined as follows:*

$$A = \begin{bmatrix} A_1 & \cdots & \cdots & A_\ell \\ A_\ell & A_1 & \ldots & A_{\ell-1} \\ \vdots & \ddots & \ddots & \vdots \\ A_2 & \ldots & A_\ell & A_1 \end{bmatrix}$$

Does there exist $e \in \mathbb{F}_q^{\ell n^*}$ *of weight wt(e)≤w such that* $A \times e^T = z$?

1.3.4 Encoding constant weight words

To transform a binary string into error vector (bijective) or encode/decode constant weight words, we have two methods: the enumerative method [Sch72] and the recursive method [Sen05]. The second method, recursive method consists in a variable length encoder. It is significantly faster than enumerative method, but the major problem is security. We are interested in the enumerative method, which is based on the following bijective application:

$$\phi_{n,t} : \left[0, \binom{n}{t}\right[\rightarrow \mathcal{W}_{n,t} := \{x \in \mathbb{F}_q^n | wt(x) = t\}$$

$$x \qquad \longmapsto \qquad (i_1, \ldots, i_t)$$

The Niederreiter cryptosystem (see 1.4.2) is applied in this application for implementation and it is as well, used to transform a binary string into error vector.

$\mathcal{W}_{n,t}$ is represented by its non-zero positions in an increasing order $0 \leq i_1 < i_2 < \ldots < i_t \leq n-1$ and length of x is $\ell = \lfloor log_2 \binom{n}{t} \rfloor$.

The inverse application is defined as follows:

$$\phi_{n,t}^{-1}: \mathcal{W}_{n,t} \quad \rightarrow \quad \left[0, \binom{n}{t}\right[$$

$$(i_1, \ldots, i_t) \quad \mapsto \binom{i_1}{1} + \binom{i_2}{2} + \ldots + \binom{i_t}{t}$$

The cost of a bijective application is $O(t\ell^2)$ binary operations. The decoding algorithm $\phi_{n,t}$ is proposed by [Sch72] as follows (Algorithm 2.1):

Algorithm 2.1 Enumerative decoding

Data $x \in \left[0, \binom{n}{t}\right]$

Result t integers $0 \leq i_1 < i_2 < \ldots < i_t \leq n-1$

j ← t

while $j > 0$ **do**

$i_j \leftarrow$ invert-binomial (x,j)

$x \leftarrow x - \binom{i_j}{j}$

$j \leftarrow j - 1$

end while

where invert-binomial (x, j) returns the integer i such that $\binom{i}{j} \leq x < \binom{i+1}{j}$

1.4 Code-based encryption schemes

1.4.1 McEliece Cryptosystem

The McEliece cryptosystem [McE78] is the first public key cryptosystem based on algebraic coding theory and based on the general decoding problem. McEliece proposed a construction based on Goppa codes.

The principal idea is to first select a linear code for which an efficient decoding algorithm is known, and then to use a trapdoor function to disguise the code as a general linear code. Though numerous computationally-intensive attacks against the scheme appear in the literature, such as [FS09], no efficient attack has been found up to now. We describe this cryptosystem as following:

Private Key

- G' a generator matrix of a binary linear C,
- S a non-singular random $k \times k$ binary matrix,
- P a random binary $n \times n$ permutation matrix.

- $\psi(.)$ a polynomial-time decoding algorithm until $\frac{d}{2}$ errors.

Public Key

- G=$SG'P$ and t integer $< \frac{d}{2}$.

Encryption

- m message with length k,
- Cryptogram c'=$mG \oplus e$, where $wt(e)$=t.

Decryption

- $wt(eP^{-1})$=t and $(mQ)G$ is a codeword,
- $mS' = \psi(cP^{-1}) = \psi((mS')G' \oplus eP^{-1})$,
- $m = (mS')S'^{-1}$.

1.4.2 Niederreiter Cryptosystem

Niederreiter cryptosystem [Nie86] defined the dual version of McEliece cryptosystem using the parity check matrix which is based on the syndrome decoding problem. The security of Niederreiter's cryptosystem and McEliece's cryptosystem are equivalent. The main difference is that instead of a generator matrix, the Niederreiter PKC uses a party check matrix only. It allows to reduce the size of the public key from $k \times n$ into $(n-k) \times n$, reduce the cryptogram from n into $n-k$.

A block of a plaintext is mapped to an error vector of desired weight by a bijective application, like $\phi_{n,t}$ (described in 1.3.3). The corresponding ciphertext is the syndrome of the error vector. The Niederreiter encryption scheme is described as follows:

Private Key

- H' a parity check matrix $(n-k \times n)$ of a binary linear C,
- P a permutation matrix $n \times n$,
- Q a invertible matrix $(n-k) \times (n-k)$ permutation matrix,
- ψ a decoding algorithm until $\frac{d}{2}$ errors.

Public Key

- $H = QH'P$ and t integer $< \frac{d}{2}$.

Encryption

- Decoding message m to error vector e with length n and $wt(e)$=t,
- Calculate $S = H^t e$, where S is cryptogram

Decryption

- Calculate $z = Q^{-1}S$,
- Compute $y = \psi(z)$,
- Calculate $e=yP$,
- Encoding e into message m.

1.4.3 Randomized McEliece Cryptosystem

Nojima et al. [NIKM08] proved formally that padding the plaintext with a random bit-string provides the semantic security against a chosen plaintext attack (IND-CPA) for the McEliece (and its dual, the Niederreiter) cryptosystems under the standard assumptions. The cryptogram of Randomized McEliece cryptosystem is:

$$c'=c\oplus e= [r\|m]G\oplus e=(rG_1\oplus e)\oplus mG_2$$

where

- $G=[G_1 \| G_2]$
- k_1 and k_2: two integers such that $k=k_1+k_2$ and $k_1<bk$ where $b<1$,
- G_1 and G_2 : matrix with k_1xn and k_2xn, respectively,
- r: random string with length k_1,
- m: message with length k_2.

The encryption algorithm only encrypts $[r \| m]$ instead of m itself. The decryption algorithm is almost the same as McEliece, the difference is that it outputs only the last k_2 bits of the decrypted string.

1.4.4 Randomized Niederreiter Cryptosystem

The randomized Niederreiter cryptosystem is based on the use of the random padding for enhancing security of the Niederreiter cryptosystem. The cryptogram of Randomized McEliece cryptosystem is:

$$S = [r|m]H = rH_1 \oplus mH_2$$

where:

- H: matrix (k,n) (public key), where $H=SH'P$
- $H^T = [H_1^T|H_2^T]$

- n_1 and n_2: two integers, such that $n=n_1 + n_2$
- H_1: matrix with $(n-k)$xn_1
- H_2 : matrix with $(n-k)$xn_2
- r : random string with length n_1 and weight $t_1 = \left\lfloor \frac{n_1 \times t}{n_1+n_2} \right\rfloor$
- m: message with length n_2, and weight $t_2 = \left\lfloor \frac{n_2 \times t}{n_1+n_2} \right\rfloor$

1.4.5 McEliece cryptosystem based on QC-MDPC codes

Quasi Cyclic-Moderate Density Parity Check (QC-MDPC) code is a linear block code with quasi-cyclic construction (see [MTSB13]) which permits to reduce the public key size.

- **Quasi-cyclic code**: An $C(n,r)$-code of length $n=\ell n_0$ is a quasi-cyclic code of order ℓ (and index n_0) if C is generated by a parity-check matrix $H = [H_{i,j}]$ where each $H_{i,j}$ is an $\ell \times \ell$ circulant matrix.
- **MDPC codes**: An $C(n,r,w)$-MDPC code is a linear code of length n and co-dimension r which stands as a parity-check matrix of row weight w.

The McEliece cryptosystem based on QC-MDPC codes works as follows:

Key Generation

Generate $C(n,r,w)$-QC-MDPC code, with $n=\ell n_0$ and $r=\ell$. Select a vector F_2^n,of row weight w uniformly at random, as the initialization factor of generating $H \in F_n^{r \times n}$. The parity check matrix H is obtained from r-1 cyclic shifts by h. The matrix has the form $H=[H_0|H_1|...|H_{n_0-1}]$, where row weight of H_i is w_i and $w = \sum_{i=0}^{n_0-1} w_i$. A generator matrix $G=(I|Q)$ can be derived from the H. Note that the public key for encryption is $G \in F_2^{(n-r)\times n}$ and the private key is H.

$$Q = \begin{pmatrix} \left(H_{n_0-1}^{-1}.H_0\right)^T \\ \left(H_{n_0-1}^{-1}.H_1\right)^T \\ \dots \\ \left(H_{n_0-1}^{-1}.H_{n_0-2}\right)^T \end{pmatrix}$$

Encryption

To encrypt the message $m \in F_2^k$, where $k=n-r$

- Randomly generate $e \in F_2^n$ of $wt(e) \leq t$.
- The ciphertext $c' \in F_2^n$ is $c'=mG \oplus e$.

Decryption

Let ψ_H a decoding algorithm equipped with the sparse parity check matrix H. To decrypt c' into m

- Compute $mG = \psi_H(mG \oplus e)$,
- Extract the plaintext m from the first k positions of mG.

We mention that the public-key generated by McEliece cryptosystem based on QC-MDPC codes is less then McEliece Goppa codes. The parameters of code that provide a level of 80 bit equivalent symmetric security are: $n_0 = 2$, $n = 9602$, $r = 4801$, $w = 90$, and $t = 84$ [MTSB13]. The public-key size in McEliece QC-MDPC codes is 0.586 KB (4801 bits), however, the public-key in McEliece cryptosystem with Goppa codes is 150 KB.

1.5 Critical attacks on the McEliece cryptosystem

In literature of attacks on McEliece cryptosystem, there are two big classes of attacks: not critical attacks, and critical attacks [Cay08]. The first one is depended on the parameters of code; we can avoid these attacks by increase the value of these parameters. A detailed overview of this class of attacks can be found in [IK01, FS09]. In this section, we detail the critical attacks.

The critical attacks discord with size parameters of code, but are based on the use of structural weaknesses of the protocol. T. Berson in [Ber97] describes three critical attacks, message-resend, related-message, and partial-message attack.

1.5.1 Message-resend attack

We suppose that the intruder intercepts the ciphertext transmitted in the network with different run:

$$c_1 = mG + e_1$$

and

$$c_2 = mG + e_2$$

where $e_1 \neq e_2$. We call this a message resend attack. In this case, it is easy for the cryptanalyste to recover m here from the system of c_i. We will only examine the case where $i = 2$: The attack is even easier if $i > 2$.

Notice that $c_1 + c_2 = e_1 + e_2 \pmod 2$.

A resend of message can be detected easily while observing the weight of Hamming of the sum of two ciphertexts. When the messages are different, the expected weight of the sum is about 512 (for the original parameters of McEliece, in general the waited weight is k). When the two messages are identical, the weight of the sum cannot exceed 100 (or in general $2t$). Heiman [Hei87] proved that the resend of message can be detected.

1.5.2 Related-message attack

This attack is generalized of message-resend attack. We suppose that two ciphertexts

$$c_1 = mG + e_1$$

and

$$c_2 = mG + e_2$$

where $m_1 \neq m_2$ and $e_1 \neq e_2$, and that the intruder knows a linear relation between the plaintexts m_1 and m_2, for example m_1+m_2. We call this a related-message attack. With these conditions, the intruder may recover the m_i. Then, we obtain

$$c_1 + c_2 = m_1G + e_1 + m_2G + e_1$$

Notice that $m_1G+m_2G = (m_1+m_2)G$, a value the intruder can calculate under the condition related-message from the known relationship and the public key. It solves then:

$$c_1 + c_2 + (m_1+ m_2)G= e_1 + e_1$$

and achieve an attack by return of messages, while using $(c_1 + c_2 + (m_1 + m_2)G)$ instead of $(c_1 + c_2)$.

1.5.3 Partial-message attack

To have a partial knowledge of the plaintext reduced in a drastic manner the cost of computation of the attacks against the McEliece cryptosystem [CS98]. For example, we are m_l and mr representing the k_l bits of left and the k_r bits remaining the plaintext m, where $k = kl + kr$ *and* $m = (m_l \| m_r)$.

Let's suppose that an intruder known m_r. Then, the difficulty to recover the plaintext unknown m_l in the McEliece cryptosystem with parameters (n, k) is equivalent to recover the plaintext with parameters (n, k_l), since:

$$c = mG + e$$
$$c = m_lG_l + m_rG_r + e$$
$$c + m_lG_l = m_rG_r + e$$
$$c' = m_lG_l + e$$

Where G_r and G_l are the k_r superior lines and k_l the other lines of G, respectively.

1.6 Other Code-based cryptographic primitives

During the last years, many code-based cryptographic primitives have been designed. Here, we present an idea of these cryptographic primitives.

1.6.1 Pseudo Random Generator

B. Fischer and J. Stern [FS96] proposed the first pseudo-random generator based on error-correcting codes. This generator is based on the fact that the greater the weight of error vectors, the exponentially greater the number of words having the same syndrome. They described an efficient pseudo random generator which can output 3500 bits/sec as compared to an RSA based generator (512 bits modulus) which outputs 1800 bits/sec.

1.6.2 Identity Based Identification Scheme

Identification schemes are main tools in various applications and online systems for preventing data access by invalid users. In 1986, Fiat and Shamir [FS86] proposed a particular scheme named zero-knowledge proof. The first designed zero-knowledge identification scheme based on hardness of the syndrome decoding problem is proposed by Stern in 1993 [Ste93]. A few years later, Véron in [Vér96] has designed a scheme with a lower communication cost. In 2010, Cayrel-Véron-El Yousfi in [CVE10] has designed a scheme which reduces this communication cost even more.

1.6.3 Hash Function

D. Augot, et al. [AFS05] have been proposed a provably collision resistant family of hash functions. The Fast Syndrome Based Hash function is based on the Merkle-Damgard design which consists in iterating a compression function. This function takes as input a word of s bits, it result is a word of length n and weight t and calculates its syndrome from a given $r \times n$ parity check matrix (with $r < s$). In 2011, Bernstein et al. [BLPS11] proposed

RFSB (Really Fast Syndrome-Based Hashing). RFSB is based on random functions, and uses the AES algorithm.

1.6.4 Signature Scheme

Kabatianskii et al. [KKK97] proposed a signature scheme based on arbitrary linear error-correcting codes. Using Niederreiter's cryptosystem, N. Courtois et al. [CFS01] proposed a signature scheme which ouputs very short signatures. The principal problem is that hash values lie in the set of syndromes and must match the syndrome of an error of weight t in order to apply the decrypting function.

1.6.5 Private-key scheme

A. K. Al Jabri in [Alj97] proposes a private-key version of McEliece cryptosystem. This new variant is based on the same concept suggested by McEliece except that erasures are used instead of errors. Such a modification allows for almost doubling the amount of added errors to the encoded vector.

1.7 Conclusion

In this chapter, we showed the main concepts of public-key cryptography and coding theory. Among applications which are applied in coding theory we cited cryptography with different schemes (signature, identification, hash function, etc.). We focused on code-based encryption schemes particularly McEliece encryption scheme and its different variants. These variants are agreed on to secure a lot of RFID authentication protocols, we will show it in this thesis.

Chapter 2

RFID Systems and their Security

2.1 Introduction

RFID technology was invented in 1948, but it was not commercialized until the 1980s. RFID systems have seen rapid development in recent years and in different areas, including space memory, computing capabilities, and security. This technology is applied in different fields, such as libraries, supply chain management, access control, etc. In the survey of RFID systems, we find two principal research topics: security and evolution of performance.

This chapter consists in defining RFID systems as well as their components, their applications, and their classification. After that, we present different security and privacy properties which are required in RFID systems. Then we show numerous possible threats in RFID systems. Finally, we present different categories of RFID authentication protocols.

2.2 RFID systems

RFID is a technology without contact with incorporates and using electromagnetic or electrostatic coupling in the radio frequency (RF) portion of the electromagnetic spectrum. It makes it possible to identify an object, person, or animal. In the last years this technology has replaced the barcode, especially in industry.

The typical RFID systems comprise of three main components: the tag (or transponder), the reader (or transceiver), and the server (or backend, data processing device). The Figure 2.1 shows components of RFID systems.

2.2.1 Components of RFID systems

2.2.1.1 RFID tags

The RFID tag consists of a microchip and a coupling element, such as an antenna, to communicate via radio frequency. The microchip has memory and can store data up to 128

Kbytes. The antenna is physically attached to the microchip and is used to draw energy from the reader to energize the tag.

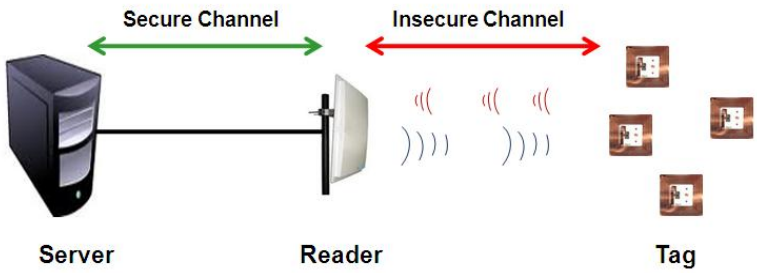

Figure 2.1: RFID Systems

2.2.1.2 RFID readers

The RFID reader is a device which communicates with tags via radio waves. It consists of one or more antennas that emit radio waves and receive signals from one or more tags. The reader sends a request as an interrogating signal for identification information to the tag. The tag responds or broadcasts with the respective information by sending an encoded modified signal, which the reader decodes, forwarding it to the server. Also, this device can be used to write data into RFID tags.

2.2.1.3 Server

The server (back-end or data processing device) is a centralized place that hosts all data regarding access permissions and may be consulted by the reader. It can provide a variety of computational functions on behalf of applications. The server provides a database of information about items identified by tags.

2.2.2 Functionality and Advantages of RFID systems

The functionality of this system is defined as follows: the RFID reader sends a signal of radio waves on a determined frequency, the tag that is in the field of action of the reader uses this signal as energy, this energy actives the chips what permits to send back the information that it contains.

The main advantages of RFID system which are related with smartcards and barcodes are:

- Tag detection does not require human intervention and thus reduces employment costs and eliminates human errors from data collection,

- Line-of-sight and direct contact are not required between the reader and the tag unlike barcode system, tag placement is less constrained,
- Wide reading range, the reader can be up to 10 meters away from the tag,
- RFID tag has a longer read range than barcode,
- Tag has read/write memory capability, while barcode do not,
- An RFID tag can store a unique identifier and also large amounts of data,
- Tag is less sensitive to adverse conditions (dust, chemicals, physical damage etc.),
- Many tags can be read simultaneously using anti-collision Identification,
- RFID tags can be combined with other devices, such as cell phone and sensors,
- RFID tags cannot be replicated easily,
- RFID system is also more stable against the vulnerablity environment factors like dirt and wearing that barcodes and optical character recognition labels face.

2.3 Classification of RFID systems

The RFID systems can be classified into different classes according to the following criteria: frequency, power source, memory, standard, and fixation of reader. These characteristics are interdepending. Figure 2.2 presents different classes of RFID systems.

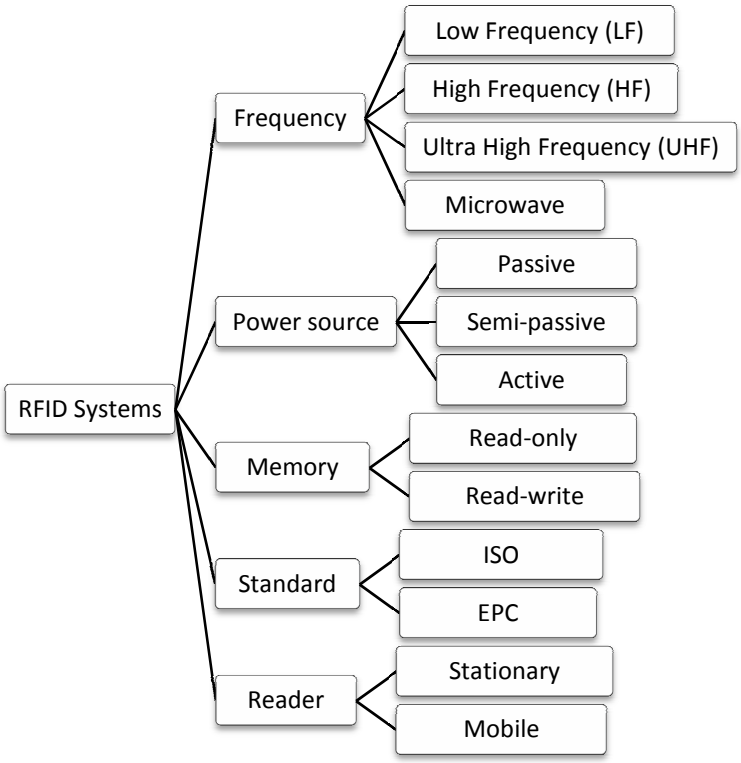

Figure 2.2: Classification of RFID systems

2.3.1 Frequency

RFID system is based on wireless communication and makes use of radio waves that are a part of the electromagnetic spectrum. It operates on different frequencies depending on the application. Generally, these operating frequencies are classified into four frequency bands. Table 2.1 shows the characteristics of each band with their respective applications.

Band	Low Frequency LF	High Frequency HF	Ultra High Frequency UHF	Microwave
RFID Frequency	125 – 134 kHz	13.56 MHz	433MHz, 860 – 960 MHz	2.45 GHz ~
Communication Range	< 0.5 m	≤ 1.5m	433MHz: ≤ 100m Other: 0.5m~ 5m	≤ 10m
Characteristics	Short range, Penetrates water but not metal, Low data rate	Mid range, Penetrates water but not metal, Reasonable data rate	Long range, Cannot penetrates water or metal, High data rate	Long range, Cannot penetrates water or metal, High data rate
Application	Access control, Car, animal ID	Smart Labels, Access & Security, Apparel	Logistics, Baggage, Electronic Toll Collection	Electronic Toll Collection

Table 2.1: Classification of RFID systems by frequency

2.3.2 Power source

The tags are classified according to the power sources as follows:

- ***Passive:*** A passive tag captures its power from the incoming RF signal of a reader. It is smaller, has lower cost, requires no periodic maintenance, and is very inexpensive.
- ***Semi-passive:*** has a battery and requires the power of reader to transmit message back to the reader. It is usually of UHF frequency band. Some semi-passive tags are in eve until they are activated by a signal coming from the reader, in order to keep the autonomy of the battery. These tags are sometimes called tags assisted by battery.
- ***Active:*** Contains a battery and sends signals automatically to the reader. It has the advantage of longer reading distance as no power has to be transmitted wirelessly. The most expensive but is typically used in logistic applications. It can be of UHF or Microwave frequency bands.

2.3.3 Memory

Another classification is based on the characteristics of the types of memory. The memory of a tag generally consists of a containing ROM (Read Only Memory), the information of security, as well as a gone resident of the operating system, and one RAM (Random Access Memory) that represents the programs executes themselves.

- ***Read only*** information on the tag is factory programmed, and the memory is disabled to prevent future changes. It is a very limited quantity of data can be stored.
- ***Read-Write*** can be read as well as written into. It contains more memory (32kB to 12kB) but it is more expensive than the read only tags.

2.3.4 Standard

Standardization was needed for the interoperability of the RFID systems from various vendors. The International Standards Organization (ISO) has created standards for air interface protocol, data content, conformance and performance testing for RFID systems. EPCglobal has designed electronic product code (EPC) system for the use of RFID technology. Standards of ISO and EPCglobal are related to physical, communication, and application layers.

- ***EPCglobal*** EPCglobal [Epc] was a GS1 (*General Specification*) initiative to develop industry-driven standards. The Electronic Product Code (EPC) is a syntax for unique identifiers assigned to physical objects, unit loads, locations, or other identifiable entities playing a role in business operations. EPCs have multiple representations, including binary forms suitable for use on RFID tags, and text forms suitable for data sharing among enterprise information systems. GS1's EPC Tag Data Standard (TDS) specifies the data format of the EPC, and provides encodings for numbering schemes within an EPC. When unique EPCs are encoded onto individual RFID tags, radio waves can be used to capture the unique identifiers at extremely high rates and at distances well in excess of 10 metres. These characteristics of RFID can be leveraged to boost supply chain visibility and increase inventory accuracy. One of the most recent standard of EPCglobal is EPC Class 1 Gen 2. It works up to a couple of meters, and it is very sophisticated in inventorying, session management, etc.

- ***ISO*** With ISO group, we can find the following norms:
    - ***ISO/IEC 14443*** This norm specifies a class of RFID proximity tags. It is used in transportation systems, building access, Visa paypass. The cards operate in the 13.56 MHz band and they have a range of a few dozen centimetres.
    - ***ISO/IEC 15693*** This norm specifies a class of RFID vicinity tags. The ISO 15693 tags operate also in the 13.56 MHz band and they have a far greater operating range which can be between 1 and 1.5 meters.
    - ***ISO/IEC 15459*** This norm defines a class of unique identifiers for transport units, including supply chain items and containers. It is roughly equivalent to the specification of different serialized of EPC. It can be represented in multiple forms: barcodes and RFID.
    - ***ISO/IEC 18000*** This norm was first published in 2004. This implicates a conflict with the EPC Gen2 specification which was developed in parallel. After that, this conflict was corrected in 2006. The norm of ISO/IEC 18000 provides the specific values for definition of the air interface parameters for a particular frequency including LF, HF, UHF, microwave and passive or active tags.

2.3.5 Fixation of readers

We have two categories of readers according to their fixation:

- ***Stationary*** The reader is attached in a fixed way, for example at the entrance gate, and respectively at the exit gate of people.
- ***Mobile*** In this case the reader is a handy, movable reader, for example in inventory management.

2.4 RFID Applications

RFID applications can be used by the individuals and the enterprises as well as by the states. There are numerous RFID applications available today, such as: transportation, animal identification, health, library, access control, etc. Figure 2.3 shows examples of RFID applications.

Library Among the important uses of RFID systems is its deployment in libraries. Use of RFID technology in libraries can facilitate lending library items (books, DVD, CD, etc.), and to tracking and tracing these ones. Moreover, the RFID tag contains identifying

information, such as a book's title or book's authors. In last years, this technology has replaced the old identification method of books, which is barcode.

Figure 2.3: Examples of RFID Applications

Access Control Contactless access control with RFID tags is popular for securing physical locations, such as office buildings, individual rooms, and commercial premises. First invented in 1973 by Charles Walton, the original RFID-based access control system involved an electronic lock that opened when presented with an RFID key card. One has two different access control systems: online and offline system. The first system tends to be used where the access authorization of a large number of people has to be checked at just a few entrances. All RFID readers are connected to a server by means of a network. The second system has become prevalent primarily in situations where many individual rooms, to which only a few people have access, are to be equipped with an electronic access control system.

e-Passports An e-Passport contains a RFID tag, This tag holds the same information that is printed on the passport's data page: the holder's name, date of birth, and other biographic information. An e-Passport also contains a biometric identifier and the travel history (date, time, and place) of entries and exits from the country. Many countries use RFID passports ("e-passport") as authentication document in transportation between countries.

Animal identification One of very useful techniques in animal identification is implantable RFID tags. This permits to identify the animal at a distance of up to 1 m,

verification of origin and the control of epidemics, measuring milk output, and automatic feeding in a feeding stall. Many options have been found for attaching the tag to animals: collar tag, injectible tag, Bolus, and ear tag.

Human implants Implantable RFID tags designed for animal tagging are now being used in humans. An early experiment with RFID implants was conducted by Kevin Warwick, professor of cybernetics, who implanted a tag in his arm in 1998. For example, The Mexican Attorney General's office in 2004 implanted as set of its staff members with the Verichip to control access to a secure data room.

Health The RFID technology is used in hospitals to identify patients and permit relevant hospital staff (e.g. physicians and nurses) to access medical records. For example, the Verichip society produces an implantable RFID tags to identify patients in emergency situation. In addition, adapting RFID technology in healthcare systems has helped hospitals in reducing medication errors.

Supply Chain Management RFID application in the supply chain offers solutions when it is impractical to use other technologies like barcode to collect data. RFID tags can be attached directly to the materials or items and they can be attached to the containers that carry them. RFID can be used to monitor and manage the movement of products at different points in the supply chain: manufacturing, warehouse, distribution and retail. RFID technology can decrease costs associated with product tracking and inventory counting. It can increase the accuracy and timelines of inventory data. Also, it is even possible to control that the products are transported in the requisite conditions while verifying the temperature for example. The important standard oriented to supply chain application is EPC RFID.

Transportation One of the most known applications and uses of the RFID technology remains the transportation industry and this in many areas: trucking, airports, rail, shipping, and tolls [Flo14]. We explain how RFID technology is used in two first areas. In trucking, the car's RFID tag would be read and the pertinent data reviewed (taxes, safety, weight, etc.). If the car met all the standards required, a signal would be sent to the car allowing the station to be bypassed. Some airports have implemented control of taxis and busses by tagging them and then checking the amount of time or number of trips each vehicle makes. By charging a fee for any excess trips or wait time, airports have been able to free up curb space and reduce congestion.

2.5 Security and privacy properties

In order to have secure authentication protocols, it is important that a RFID authentication protocol requires security and privacy proprieties.

- ***Secrecy*** or confidentiality, keeping tag's identifier, cryptographic keys or other secret information from all but the server and the tag. This secret information is never passed on clearly to air on the radio frequency interface which can be spied on.
- ***Integrity*** If an adversary modifies data of a legitimate tag while the data are being in transit, the reader should be able to detect this modification. To detect this modification, there are several techniques, like hash function, MAC (*Message Authentication Code*), and digital signature.
- ***Mutual authentication*** A RFID authentication scheme achieves mutual authentication, that is to say, it achieves reader's authentication and the tag's authentication:
    - ***Tag authentication*** A reader has to be capable of verifying a correct tag to authenticate and to identify a tag in complete safety.
    - ***Server authentication*** A tag has to be capable of confirming that it communicates with the legitimate reader (a single reader exists in communications between the constituents of the RFID system).
- ***Untraceability*** The untraceability is one of privacy proprieties. An RFID system satisfies untraceability if an intruder cannot find any links among any readings of the same tag. This implies that the intruder cannot track of the tag as run in different sessions. This property is called also location privacy.
- ***Desynchronization resilience*** We can define this property as follows: at session (*i*), the intruder can modify or block the transmitted messages between the tag and the reader. In the next session, if the authentication process fails, then the tag and the reader are not correlated and this protocol does not achieve desynchronization resilience. We note that this property specifies for the RFID protocols that update a shared secret before terminate the protocol.
- ***Forward secrecy*** One of the abilities of the intruder, is to compromise the secrets stored in the tag. The property of forward secrecy signifies protecting the previous communications from a tag even when assuming that the tag has been compromised.

- ***Non-repudiation*** Prevents a user (tag or reader) from denying previous commitments or actions. Concerning RFID, the threats of repudiation occur when a user refuses an action and no proof exists to confirm that the action has been achieved.

2.6 RFID Threats

The communication channel between the server and the reader is assumed to be secure while the wireless channel between the reader and the tag is insecure since it makes it open to attacks on RFID system. We assume that the intruder has a complete control over the channel of communication between the reader and the tag. It can intercept any message passing through the network, modify or block messages, and it can also create new messages from its initial knowledge. These used assumptions are gathered under the name Dolev-Yao model [DY83]. One of the most important studies on RFID threats is [MRT08].

- ***Tag Tracing*** It consists in tracing a tag and thus, a customer in space or time. The goal of the intruder is to trace a tag.
- ***Replay attack*** Replay attack is an impersonation attack where the intruder replays or resends previous transmitted messages between reader and tag in the same session or in various sessions of same the protocol to be authenticated as legitimate reader or tag.
- ***Man in the middle attack (MITM)*** The intruder could interfere with messages exchanged between a reader and a tag by modification, insertion, or deletion, in order to impersonate it later.
- ***Relay Attack*** In a relay attack an intruder acts as a MITM. An intruder device is placed surreptitiously between a legitimate reader and the tag to intercept the communications between the reader and the tag.
- ***Denial of Service attack*** The RFID system is in regular work if the tag and the server are available. The system does not resist denial of service attack if the intruder can block RFID readers' signals or realizes desynchronization between the tag and server, i.e. the intruder can block or modify the messages transmitted between the reader and the tag so that they are not correlated in future authentication sessions.
- ***Eavesdropping*** The intruder can eavesdrop because the communication between the reader and the tag is wireless and based on radio frequency.
- ***Tag Cloning*** An intruder can read the legitimate tag, after that clone the legitimate tag by writing all the obtained data into a rogue tag. Cloning does not just mean

copying a tag's identification and data but creating a new tag that follows the original one even to the form factor.

2.7 Privacy model

In the literature of verification of privacy properties, we can find many privacy models. One of the first privacy models was proposed by Avoine et al. [ADO06] which is based on the notion of indistinguishability. Juels and Weis [JW07] extended this model using side-channel information and making the two target tags chosen by the intruder. Another model was proposed by Ouafi and Phan [Oua12, OP08] which is based on the Juels-Weis model. Authors added numerous definitions in the untraceability property.

Ouafi and Phan capture the notion of privacy as the inability for any adversary to infer the identity of a tag chosen from a pair he has chosen. After interacting with the RFID system, the adversary is asked to select two RFID tags and receives one of them. The main goal is to discover the identity of the received tag. For that, it is still allowed to interact with the system and the target tag. The adversary has defeated the privacy of the scheme if he guesses for the correct identity of the true tag with a probability significantly greater than the one of output as a random guess.

Now, we present the formal definition of Ouafi and Phan model, protocol party is a tag $T \in Tags$ or a reader $R \in Readers$ interacting in protocol sessions as per the protocol specifications until the end of the session. An adversary $\mathcal{A}$ is a malicious entity, modeled as a probabilistic polynomial-time algorithm, which controls the communications between readers and tags and interacts with them as defined by the protocol. The adversary is allowed to run the following queries:

- ***Execute (R,T,i) query***. This query models the passive attacks. The adversary $\mathcal{A}$ eavesdrops the communication channel between T and R and gets reading access to the exchanged messages in session i of a truthful protocol execution.
- ***Send (U,V,m,i) query***. This query models active attacks by allowing the adversary $\mathcal{A}$ to impersonate some reader $U \in Readers$ (respectively tag $V \in Tags$) in some protocol session i and sends a message m of its choice to an instance of some tag $V \in Tags$ (respectively reader $U \in Readers$). Furthermore the adversary $\mathcal{A}$ is allowed to block or alert the message m that is sent from U to V (respectively V to U) in session i of a truthful protocol execution.

- ***Corrupt (T,K') query***. This query allows the adversary $\mathcal{A}$ to learn the stored secret *K* of the tag *T* ∈ *Tags*, and which further sets the stored secret to *K'*. Corrupt query means that the adversary has physical access to the tag, i.e., the adversary can read and tamper with the tag's permanent memory.
- ***Test*** (i, T_0, T_1) ***query***. This query does not correspond to any of $\mathcal{A}$'s abilities, but it is necessary to define the untraceability test. When this query is invoked for session *i*, a random bit $b \in \{0,1\}$ is generated and then, *A* is given $T_b \in (T_0, T_1)$ Informally, $\mathcal{A}$ wins if he can guess the bit *b*.

Definition 2.1 (Freshness)

A party instance is fresh at the end of execution if, and only if,

- *it has output* Accept *with or without a partner instance,*
- *both the instance and its partner instance (if such a partner exists) have not been sent a* Corrupt *query*

Definition 2.2 (Untraceable privacy (UPriv))

Untraceable privacy is defined using the game played between an adversary $\mathcal{A}$ and a collection of the reader and the tag's instances. This game is divided into three phases:

- ***Learning phase:*** *$\mathcal{A}$ is able to send any* Execute, Send*, and* Corrupt *queries at will.*
- ***Challenge phase:*** *$\mathcal{A}$ chooses two fresh tags T_0, T_1 to be tested and sends a Test query corresponding to the test session. Depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}$ is given a tag T_b from the set $\{T_0, T_1\}$. $\mathcal{A}$ continues making any* Execute*, and* Send *queries at will.*
- ***Guess phase:*** *finally, $\mathcal{A}$ terminates the game and outputs a bit $b' \in \{0, 1\}$, which is its guess of the value of b.*

The success of $\mathcal{A}$ in winning the game and thus breaking the notion of UPriv is quantified in terms of $\mathcal{A}$'s advantage in distinguishing whether $\mathcal{A}$ received T_0 or T_1, in other term, it correctly guessing b. and denoted by $Adv_{\mathcal{A}}^{UPriv}(k)$ where k is the security parameter.

2.8 Classification of RFID authentication protocols

In the classification of the authentication protocols in RFID systems, we can find several factors. We cite two important classifications, by stating of shared secret and required primitives.

2.8.1 State of shared secret

In the protocols using secret shared (tag's identifier, symmetric-key, etc.), two mechanisms are used: static and dynamic. The characteristic of the mechanism of static secret is that the shared secret remains the same during the complete authentication, but that of the dynamic mechanism, the shared secret is modified. We cite example of RFID authentication protocols with dynamic shared secret, like [LAK07, Chi13].

2.8.2 Required primitives

This classification is based on cryptographic and algebraic primitives which are used in authentication protocols to assure the security and privacy properties. We mention that these classes of primitives are as follows: public-key cryptosystem, private-key cryptosystem, hash function, lightweight function, and bitwise operators (see Table 2.2). All these classes except the last class require a PRNG (Pseudo-Random Number Generator) for generating nonces. They are used to avoid replay attacks. The difference between these classes lies in the realized security properties and the complexity of implementation.

2.8.2.1 Public-key cryptosystem

Public-key cryptosystem is divided into three families according to the hardness problem: cryptosystem based on number theory, public-key cryptosystem based on coding theory, and cryptosystem based on lattice.

Public-key cryptosystem based on number theory The majority of RFID authentication protocols which require these cryptosystems use ECC [Mil85] (Elliptic Curve Cryptosystem) cryptosystem (e.g. [HKCL14]) and avoid to use the RSA and ElGamel cryptosystems. The advantage of ECC compared with RSA and ElGamel is the smaller key sizes and compatibility with available resources of RFID tags. A key size of 190 bit for an ECC is approximately equivalent to an RSA key size of 1937 bit. Concerning the implementation, ECC requires less gates compared to RSA, ECC-256 is

possible with less than 10000 GE (gates equivalents), whereas RSA needs about 50000 GE. This cryptography is used in narrow domains, like e-passport.

Class	Sub-class	Scheme	Examples of RFID protocols
Public-key cryptosystem (PKC)	PKC based on number theory	ECC	[HKCL14, KGA15]
	PKC based on coding theory	McEliece and its variants	[MM12, Chi13, LYL14]
		Niederreiter and its variants	[Cui07, SKI10]
	PKC based on lattice	NTRU	[EL12, XPK14]
Private-key cryptosystem	Block cipher	AES	[FDW04]
	Stream cipher	A2U2	[DRL11]
Hash function	-	-	[LAK06, Liu08, WHC11, JDTL12, Khe14]
Bitwise operators	-	-	[PCMA06, Chi07, Zen09]

Table 2.2: Classification of RFID authentication protocols

Public-key cryptosystem based on lattice NTRU cryptosystem [HPS98] is the most practical lattice-based encryption scheme known. The NTRU cryptosystem is required in various RFID protocols, like [EL12, XPK14]. Its faster key generation and less memory usage allow it to be used in embedded devices, like smart-cards and RFID tags. To implement this cryptosystem, one requires 3000 GE.

Public-key cryptosystem based on coding theory In this class, there are numerous RFID authentication protocols that use different code-based cryptosystems, such as [Cui07, SKI10, MM12, Chi13, LYL14]. These cryptosystems are McEliece and Niederreiter cryptosystems and theirs proposed variants. The tag (except some protocols, such as [Cui07, SKI10]) does not require a public matrix or other matrices, but it stores the codeword with the necessary information in the tag's memory. It needs a PRNG to generate an error vector and bitwise operators to compute the ciphertext.

2.8.2.2 Private-key cryptosystem

Feldhofer et al. [FDW04] proposed a first RFID protocol based on AES cryptosystem. They proposed two variants: unidirectional and mutual protocol. They also implemented this cryptosystem in RFID tag while using about 3400 GE, with a maximal clock frequency estimated to 80MHz, the consumption of energy 8.2 µ TO @ 100kHz and the maximal debit 9.9 Mbps. David et al. [DRL11] proposed a stream cipher for RFID,

called A2U2. It provides high throughput (1 bit per clock cycle) and requires very less number of logical gates, 284 GE.

2.8.2.3 Hash Function

In the survey about the design of RFID protocols, we found an important number of protocols which require a hash function, such as [LAK06, Liu08, WHC11, JDTL12, Khe14]. This primitive is a mechanism that can be integrated with message authentication code (MAC) or digital signature.

The complexities of the cryptographic hash functions standards in the integrated circuits of the type ASIC (Application-Specific Integrated Circuit) are: Fast SHA-256 and the need of about 23.000 GEs (with a maximal clock frequency estimated at 150MHz and the debit 1163 Mbps). Guo *et al.* [GPP11] designed the Photon hash-function, it has various instances (80, 128,160, and 256) and it is strong against differential and linear cryptanalysis. Photon requires lesser number of GE, e.g. Photon-80 requires only 865 GEs.

2.8.2.4 Bitwise operators

This class needs only the bitwise operators, such as AND, OR, XOR, etc. These operators are used in an important number of RFID authentication protocols, like EMAP protocol [PCMA06] and SASI protocol [Chi07], and various variants of HB protocol [Zen09]. One can implement these operators with a limited number of logical gates.

2.9 Conclusion

In this chapter, we have presented different concepts of RFID systems: definition, classification, and applications. In addition, we have showed the main notions of RFID security: security and privacy requirements, classification of RFID authentication protocols, threats, and privacy model. We presented with detail the privacy model which is proposed by Ouafi and Phan.

The bitwise operators are used in most of RFID authentication protocols for low-cost RFID tags beside other cryptographic primitives. In spite of the importance of this primitive, the abuse of bitwise operator in the exchanged messages implicates an important attack that is algebraic replay attack (ARA). In the next chapter, we detail this attack with one of bitwise operators which is or-exclusive.

Chapter 3

Algebraic Replay Attacks

3.1 Introduction

Among the attacks studied in the last years by researchers, we cite algebraic replay attacks (ARA). The main cause of these attacks is the abuse of the algebraic operator properties employed by the protocols. The operator or-exclusive (xor) is an algebraic operator. This operation is used in many RFID authentication protocols and has aroused a lot of interest during the last years; its implementation is low cost and requires few logical gates.

In this chapter, we analyse different recent RFID protocols, the common characteristic between the studied protocols are: (i) they use or-exclusive operator and one-way function in transmitted messages and (ii) the vulnerabilities of these protocols are of type algebraic replay attacks on authentication (ARA).

This chapter is based on our works [CCB12b, CCB13], it is articulated around the verification of RFID authentication protocols by using the AVISPA tools [ABB+05] after specifying these protocols in HLPSL (High-Level Protocol Specification Language) language [Tea06]. These analyses are based on the automatic verification of three security proprieties: secrecy, tag authentication and server authentication. We check which of the presented protocols cannot resist algebraic replay attacks.

3.2 Formal Automatic Verification

To verify the cryptographic protocol, we use a formal tool of verification. There are several tools of automated verification of protocols such as [KW96, Son99, GK00, ABB+05]. We select AVISPA tools (Automated Validation of Internet Security Protocols and Applications) [ABB+05] for the following reasons:

- The four available tools use various techniques of validation: Model-checking, automate trees, Solver SAT and resolution of constraints.
- Among the four tools, two tools are employed OFMC (On-the-fly Model-Checker) and CL-ATSE (Constraint-Logic-based Attack Searcher), they can verify the protocols using algebraic properties of XOR (exclusive-or) and modular exponentiation.
- The AVISPA platform is the analyzer which models a big number of cryptographic protocols (more than 90 protocols).
- These tools are based on only one specification language named HLPSL language.
- AVISPA tools can detect passive and active attacks, like replay and man-in-the-middle attacks.

Glouche et al. [GGH+09] developed a SPAN (Security Protocol ANimator) tool to animate the security protocols which are specified by HLPSL and verified by AVISPA tools. The SPAN tool permits to simulate a protocol, intruder and scenarios of attacks.

The formal automatic verification of cryptographic protocols involves the following steps:

- Specification: specification of the initial assumptions, the capacity of intruder, the protocol goals (secrecy, authentication, etc.), the roles (the tag and reader), the messages transmitted and the primitives (hash function, PRNG, xor-operator, concatenation, etc.),
- Verification: After verifying the protocol using a validation tool, it is confirmed if the protocol is either safe or it has failed. In case of failure, the tool presents the message transmitted between the intruder, reader and tag, i.e. describes the trace of attack.

3.2.1 Intruder Model

Beside modelling security protocols, it is also necessary to model the intruder, that is to say, to define its behaviour and limit. For this, we assume an active Dolev-Yao attacker [DY83]. This intruder model is based on two important assumptions that are the *perfect encryption* and the *intruder is the network*.

Perfect encryption ensures in particular that: *(1)* an intruder can decrypt a message *m* encrypted with key *k* if it has the opposite of that key, *(2)* a key cannot be guessed (during

the period of its validity), *(3)* and Given *m*, it is not possible to find the corresponding ciphertext for any message containing *m* without knowledge of the key.

The intruder is the network The intruder has complete control over the network, i.e. it can impersonate a tag, impersonate a reader, obtain any message passing through the network, block or modify messages and it can also derive new messages from its initial knowledge and the messages that are received from honest participants during protocol run. The communication between the tag and reader is not assured as it is based on radio frequencies waves. Our particular verification gets transmissions on the canal reader-tag only.

For the security protocols that require or-exclusive operator, there is another important assumption, an intruder that can exploit the algebraic properties of the XOR operator, which are:

$x \oplus 0 \rightarrow x$ (neutral element) (1)

$x \oplus x \rightarrow 0$ (nilpotence) (2)

$x \oplus y \rightarrow y \oplus x$ (commutativity) (3)

$x \oplus (y \oplus z) \rightarrow (x \oplus y) \oplus z$ (associativity) (4)

3.2.2 Specification

AVISPA provides a language called the *High Level Protocol Specification Language* (HLPSL) [Tea06] for describing security protocols and specifying their intended security properties, as well as a set of tools to formally validate them.

High Level Protocol Specification Language (HLPSL) is a modular, expressive, formal, role-based language. The HLPSL specification of protocol consists of two parties: basic roles and composition roles. The first part presents honest participants and the second part describes scenarios of basic roles.

Composition roles consist of: session, environment and goal. The session role defines the initial state of the system. The environment role shows sessions of protocol between honest participants. Before terminating the specification, we determine the security properties that we want to verify. HLPSL can specify the secrecy and the authentication

properties. Figure 3.1 shows the structure of HLPSL specification of cryptographic protocol.

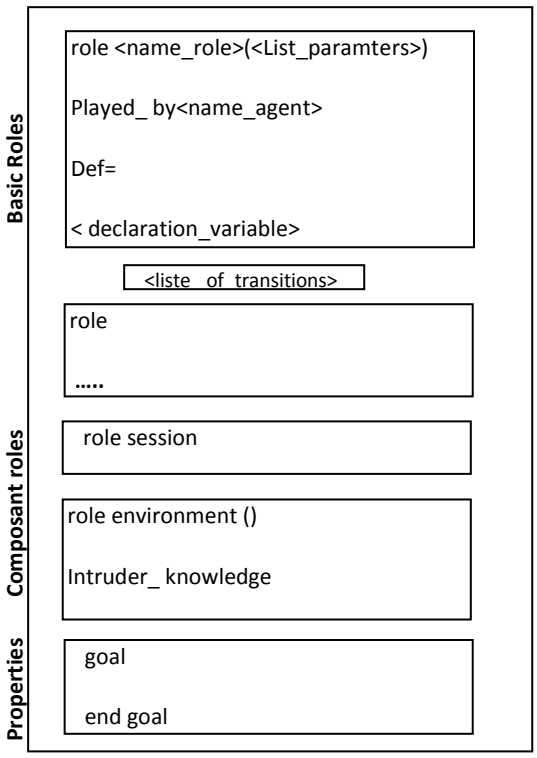

Figure 3.1: Structure of HLPSL specification of protocol

3.2.3 Verification Tools

AVISPA European Project developed four tools: On-the-fly Model-Checker (OFMC), Constraint-Logic-based Attack Searcher (CL-ATSE), SAT-based Model-Checker (SATMC), and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). OFMC and CL-ATSE can verify the protocols requiring the operator exclusive or (XOR). The architecture of AVISPA is shown in Figure 3.2.

OFMC consists of two modules. The classical module performs verification for a bounded number of transitions of honest agents using a constraint-based representation of the intruder behavior. The fixed point module allows verification without restricting the number of steps by working on an over-approximation of the search space that is specified by a set of Horn clauses using abstract interpretation techniques and counterexample-based refinement of abstractions. Running both modules in parallel, OFMC stops as soon as the

classic module has found an attack or the fixed point module has verified the specification, so as soon as there is a definitive result.

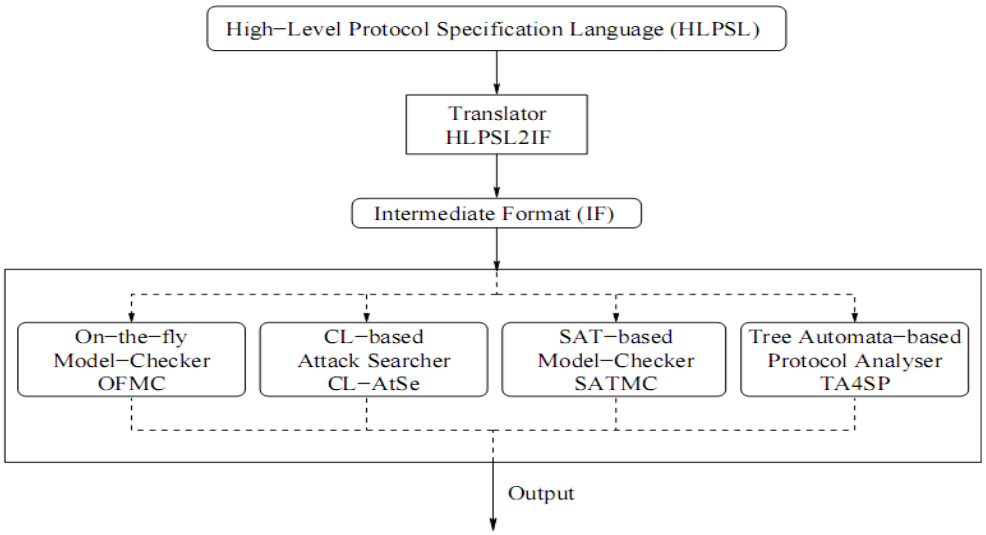

Figure 3.2: Architecture of the AVISPA Tools [ABB+05]

CL-ATSE is a Constraint Logic based Attack Searcher for the security protocols and services. It takes as an input a service specified as a set of rewriting rules, and applies rewriting and constraint solving techniques to model all states that are reachable by the participants and decides if an attack exists with respect to the Dolev-Yao intruder.

The SAT-based Model-Checker (SATMC) builds a propositional formula encoding a bounded unrolling of the transition relation specified by the IF, the initial state and the set of states representing a violation of the security properties. The propositional formula is then fed to a state-of-the-art SAT solver and any model found is translated back into an attack. SATMC does an analysis with a finished number of sessions where the messages exchanged on the network are controlled by Dolev-Yao intruder.

The TA4SP tool computes either an over-approximation or an under-approximation of the intruder knowledge by means of rewriting on tree languages in a context of unbounded number of sessions. The TA4SP tool uses the tree automata library Timbuk 2.0 to perform the computation of the intruder knowledge (over or under approximated).

3.3 RFID Authentication Protocols

We can describe the transmitted messages in studied RFID mutual authentication protocols in the form:

```
R → T : Nr
T → R : Nt, Auth_Tag
R → T : Auth_Reader
```

Protocol	**Auth_Tag**	**Auth_Reader**	**Secret Data**	**α**	***f***
Lee et al. [LAK06]	H(Nr⊕Nt⊕K)	H(H(Nr⊕Nt⊕K)⊕K⊕Nr)	K	K	H
Chien and Huang [CH07]	LH(RH(id)⊕h(Nr⊕Nt⊕K))	RH(RH(id)⊕h(Nr⊕Nt⊕K))	id, K	K	H
Liu [Liu08]	x⊕h(h(K)⊕nt), h(y⊕Nr⊕ Nt)	y^*⊕h(x^*⊕y), h(x^*⊕y^*)	h(k),y,x,K	y	H
Qingling et al. [QYY08]	CRC(id⊕Nt⊕Nr), CRC(id⊕ Nt⊕Nr) ⊕ x	CRC(id⊕Nt), CRC(id⊕Nt) ⊕ x	id	id	CRC
Wei et al. [WHC11]	H(Nr⊕Nt⊕S)	H(id⊕N_{db})	S, id	S	H
Jialiang et al. [JDTL12]	H(Nr⊕Nt)⊕S	H(Nr⊕Nt⊕N_{db}) ⊕ id, N_{db}	S, id	id	H

Table 3.1: RFID Authentication Protocols

The transmitted messages of *Auth_Tag* and *Auth_Reader* are presented in Table 3.1 The authentication message comprises $f(\alpha \oplus N_t \oplus N_r)$, with α as secret data shared between the tag and reader and f as one-way function like hash function and CRC function. The exception case is JDTL, where the message is $f(N_t \oplus N_r) \oplus \alpha$. The following is a detailed description of each step of these protocols:

- The reader RFID produces a nonce Nr and sends it and a request to the tag.
- After receiving Nr, a tag generates a random number Nt and computes the function *Auth_Tag*, then sends Nt. The *Auth_Tag* is back to the reader (server).
- After receiving the authentication message from the tag, the reader would search whether there exists certain ά in table α of the database, which could make f(ά

⊕Nt⊕Nr)= f(α⊕Nt⊕Nr). If it is found, the tag crosses the authentication of the tag and is considered as legitimate, and then the reader calculates *Auth_Reader*, then sends *Auth_Reader* to the tag.

- The tag computes *Auth_Reader*', if the outcome equals the received *Auth_Reader*, the authentication of the reader is successful; otherwise, the authentication has failed.

In our chapter, we verify six protocols, as follows (see Table 3.1):

- Lee et al. [LAK06]: Lee et al. propose an authentication protocol. The reader R and tag T share secret k. At finish authentication, reader and tag updates k to $h(k)$.
- Chien et al. [CH07]: The CH protocol was proposed by Chien and Huang in 2008. It uses hash function and non-cryptographic primitives (Left, Right and Rotate). It uses these primitives to increase the security of protocol.
- Liu [L08]: The author Yanfei Liu provided a detailed security analysis of the protocol and claimed that YL achieves a list of security properties, including resistance to tag impersonating, denial of service, replay and compromising attacks.
- Qingling et al. [QYY08]: The authors of this protocol claim that this protocol is secure because of the use of CRC (Cyclic Redundancy Check) and use of random nonces to encrypt messages.

In the next sections, we verify two recent protocols; the first protocol is proposed by Wei et al. [WHC11] and the second is proposed by Jialiang et al. [JDTL12].

3.4 Wei et al.'s Protocol

3.4.1 Review of Wei et al.'s Protocol

Wei et al. [WHC11] proposed an authentication protocol where the reader R and tag T share secrets value s and Identifier id. Figure 3.3 shows the process of the WHC protocol.

The following is a detailed description of each step of this protocol:

- The reader generates a random number N_R and query tags with N_R.
- After receiving N_R, the tag generates a random number N_T and calculates $h(s \oplus N_R \oplus N_T)$, then sends N_T and $h(s \oplus N_R \oplus N_T)$ back to the reader.
- After receiving N_T and $h(s \oplus N_R \oplus N_T)$ from the tag, the reader calculates $h(\text{RID} \oplus N_R)$, and sends N_R, $h(s \oplus N_R \oplus N_T)$, N_T, $h(\text{RID} \oplus N_R)$ to the server.

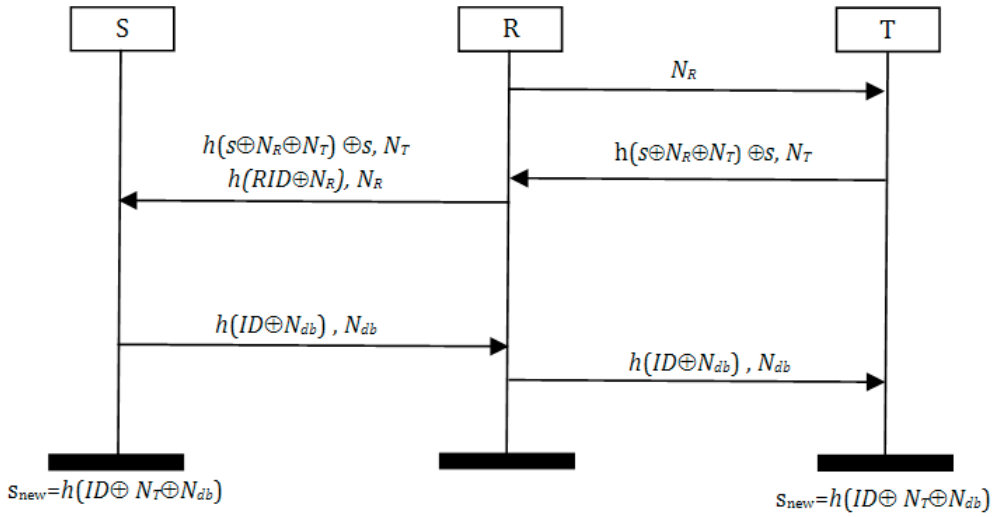

Figure 3.3: The Wei et al.'s protocol [WHC11]

- After receiving an authentication message from the reader, the server checks whether N_R matches with $N_{R(old)}$, if they match, the authentication is succeed. If they don't match, the authentication is failed.
- The server would check whether there exists certain *RID** in table *RID* of the database, which could make $h(RID^* \oplus N_R) = h(\text{RID} \oplus N_R)$. If there exists such a record, the authentication application would be considered as from a legitimate reader, otherwise authentication is failed.
- Subsequently, the server would check whether there exists a certain *s** in table *ID* of the database, which could make $h(s^* \oplus N_R \oplus N_T) = h(s \oplus N_R \oplus N_T)$. If there exists such a record, the tag would be considered as a legitimate tag, then the server generates a random number N_{db} and calculates $h(id \oplus N_{db})$, then sends N_{db} , $h(id \oplus N_{db})$ to the reader, subsequently the server should update $N_{R(old)}$, $N_{R(new)}$, s_{old} and s_{new}.
- After receiving N_{db}, $h(id \oplus N_R)$ from the server, the reader would send N_{db},$h(id \oplus N_{db})$ to the tag.
- After receiving N_{db}, $h(id \oplus N_{db})$ from the reader, the tag would calculate $h(\text{id} \oplus N_{db})$, If the outcome equals the received $h(\text{id} \oplus N_{db})$, then the object of mutual authentication is achieved, the tag should update $s = h(\text{id} \oplus N_{db} \oplus N_T)$, otherwise, the authentication is failed.

3.4.2 Specification of Wei et al.'s Protocol

The Wei et al.'s protocol used the primitives: hash function, nonce and xor-operator. These primitives are supported in HLPSL. We now present the role of reader in HLPSL specification:

```
role reader ( R,T: agent, ID,RID, S: text, H : hash_func, Snd,Rec:
channel(dy))
    played_by R
    def=
    local  State    : nat,
    Nr, Nt, Ndb  : text
    init State := 0
    transition
    1. State = 0  /\ Rec(start)   =|> State' := 1 /\ Nr' := new()
                                     /\ Snd(Nr')
    2. State = 1  /\ Rec(H(xor(xor(S,Nr),Nt')).Nt')
    =|> State' := 2   /\ Ndb' := new() /\ Snd(H(xor(ID,Ndb')).Ndb')
                      /\ secret(ID,sec_id,{R,T})
    /\ request(R,T,aut_tag,Nt') /\ witness(R,T,aut_reader,Ndb')
end role
```

This role is known as *reader*, with parameters *R* and *T* of type agent, *id* and *RID* of type text, and *H* of type hash function. The *RCV* and *SND* parameters are of channel type, indicating that these are channels upon which the agent is playing the role of the reader which will communicate. The attribute to the channel type, in this case (*dy*), denotes *the intruder model* to be considered for this channel.

The parameter *R* appears in the *played_by* section, which means, intuitively, that R denotes the name of the agent which plays the role *reader*. Also note the local section which declares local variables of reader: *State* which is a *nat* (a natural number) and random numbers of type text, Nr, Nt,and Ndb . The local *State* variable is initialised to 0 in the init section.

Concerning the transition party, the first transition of the role *reader* signifies: if the value of State is 0 and the message in the channel *REC* is *start* then: Nr takes a new random value sent on channel *SND*. The goal fact *witness(R,T,aut_server,Nr')* should be read "agent *R* asserts that we want to be the peer of agent *T*, agreeing on the value *Nr'* in an authentication effort identified by the protocol *id aut_server*. "

For the second transition, if the value of *State* is 1 and the message *H(xor(xor(Nr,Nt'),S)).Nt'* on *REC* channel then the variable *State* is set to 2, and reader sends the message *H(xor(id,Ndb').Ndb'* on channel *SND*. For the predicate *secret* it signifies "the new value stored in S is a secret to be shared only between the R and T agents". The predicate *request (R,T,aut_tag,Nt')* should be read, "agent *R* accepts the value *Nt'* and now relies on the guarantee that agent *T* exists and agrees with it on this value".

```
role session(R,T : agent,ID,RID,S : text, H: hash_func)
def=
  local Sa,Ra,Sb,Rb : channel(dy)
    composition
    reader(R,T,ID,RID,S,H,Sa,Ra) /\  tag(T,R,ID,RID,S,H,Sb,Rb)
```

In the role *session*, one usually declares all the channels used by the basic roles. The channel type takes an additional attribute, in parentheses, which specifies the intruder model one assumes for that channel. Here, the type declaration channel (dy) stands for the intruder model of Dolev and Yao [DY83]. So, *reader* and *tag* can send and receive on whichever channel they want; when the intruder is the network then the intended connection between certain channel variables is irrelevant. In our specification, the *reader* sends on *Sa* some messages to *tag* which receives them on *Rb*.

```
role environment() def=
const r,t : agent,
      id,rid,s,id,s: text,
      h: hash_func,
      aut_reader, aut_tag, sec_id : protocol_id
      intruder_knowledge = {r,t,h}
    composition
    session(r,t,id,rid,s,h)
    /\  session(r,t,id,rid,s,h)
```

The role *environment* (or top-level role) contains global constants and a composition of one or more sessions, where the intruder may play some roles as a legitimate user. There is also a statement which describes what knowledge the intruder initially has, names of all agents (r and t) and hash function h. Specification of this role depends on the treatment of two identical sessions between the same tag and the same reader (T and R). This scenario allows discovering the attacks of the type replay attack

```
goal
 secrecy_of sec_id
authentication_on aut_tag
authentication_on aut_reader
end goal

environment()
```

Security goals are specified in HLPSL by augmenting the transitions of basic roles with so-called *goal facts*. We provide a validation of properties: the secrecy of tag's identifier (*sec_id*), the tag's authentication (*aut_ tag*), and the reader's authentication (*aut_reader*).

The complete HLPSL specification of Wei et al.'s protocol shown in appendix *A*.

3.4.3 Result of verification

AVISPA tools detect trace of attack on tag authentication. Figure 3.4 shows the trace of attack on WHC protocol with the CL-Atse back-end. In this trace result, i represents the intruder, (*r*, 3) the reader, and (*t*,4) the tag. The posted information such as: *n1*(*Nr*) is instance of the nonce *Nr*. *X2400* is a variable related to the internal workings of the CL-Atse back-end (in this trace is instance of the nonce *Nr*). *N5*(*Nt*) is instance of the nonce Nt.

We symbolize: n1(Nr) by N_R, X1632 by N_R', and n5(Nt) by N_T. Several comments can be drawn from the trace:

- Msg1: The reader generates a nonce N_R and the intruder captures and stores the nonce in the course of the communication.
- Msg2: The intruder generates another nonce N_R' and sends it to the tag.
- Msg3: The tag generates an instance of the nonce N_T and sends it with the hash function $h(N_R' \oplus N_T \oplus s)$ to the intruder.
- Msg4: The intruder returns the received function to the reader with $N_R' \oplus N_R \oplus N_T$.
- Msg5: The reader sends the message $h(id \oplus N_{db})$, N_{db} to the tag. This message does not depend on the discovered attack.

The attack on tag authentication is realised in Msg4. We will describe the principle of this attack in section 3.6.

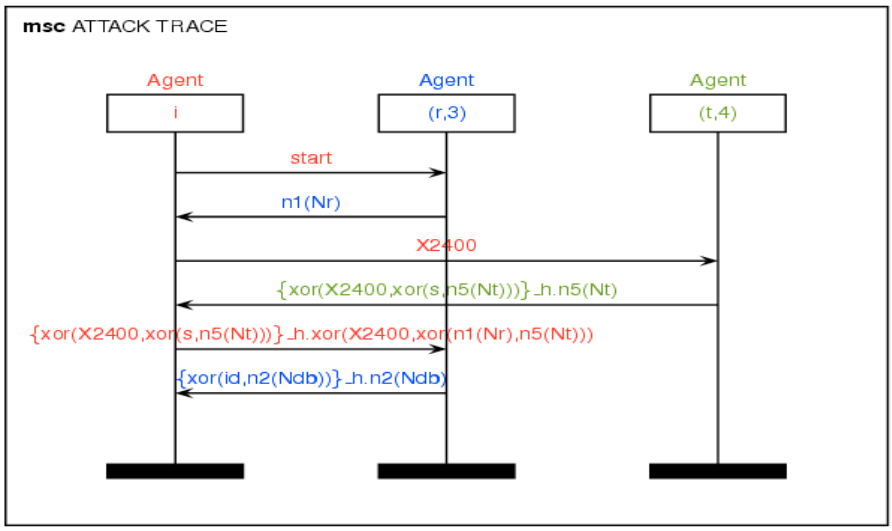

Figure 3.4: Trace attack on the WHC protocol

3.5 Jialiang et al.'s Protocol

3.5.1 Review of Jialiang et al.'s protocol

The protocol proposed by Jialiang et *al.* [JDTL12] requires hash function and PRNG. Figure 3.5 shows the process of this protocol.

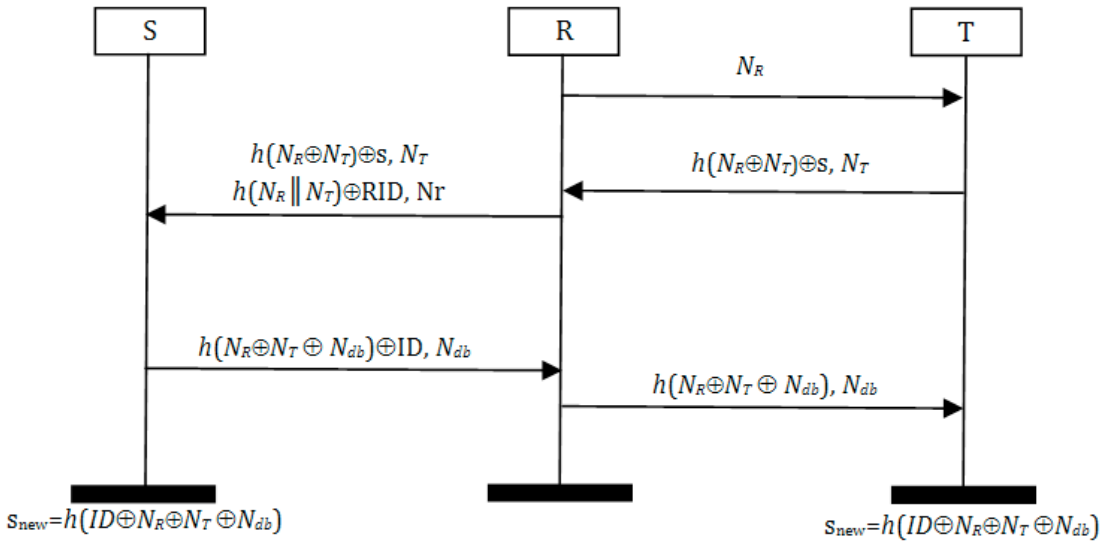

Figure 3.5: Jialiang et al.'s protocol [JDTL12]

The (i+1)*th* authentication access as follows:

- The reader generates a random number N_R and query tags with N_R.
- After receiving N_R, tag generates a random number N_T and calculates $h(N_R \oplus Nt) \oplus S$, then sends N_T and $h(N_R \oplus N_T) \oplus s$ back to the reader.

- After receiving Nt and $h(N_R \oplus N_T) \oplus S$ from the tag, the reader calculates $h(N_R \| N_T) \oplus RID$, and sends N_R, $h(N_R \oplus N_T) \oplus S$, Nt, $h(N_R \| N_T) \oplus RID$ to the server.
- After receiving an authentication message from the reader, the server checks whether N_R matches with $N_{R(old)}$, if they match, the authentication is failed. If they don't match,
- The server would calculate $RID' = h(N_R \| N_T) \oplus (h(N_R \| N_T) \oplus RID)$ and search wether there exists a certain RID^* in table *RID* of the database, which could make $RID' = RID^*$. If there exists such a record, the authentication application would be considered as from a legitimate reader, or authentication is failed.
- Subsequently, the server would calculate $s' = h(N_R \| N_T) \oplus (h(N_R \| N_T) \oplus s)$ whether there exists a certain s_{new}^* in table *ID* of the database, thus $s' = s_{new}^*$. If there exists such a record, the tag would be considered as a legitimate tag, then the server generates a random number N_{db} that could make the value which equals to $h(id \oplus N_R \oplus N_T \oplus N_{db})$ could not be found in column s_{old} and column s_{new}, and calculate $h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus id$, then sends N_{db}, $h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus id$ to the reader, subsequently the server should update $N_{R(old)}$, $N_{R(new)}$, s_{old} and s_{new}.
- After receiving N_{db}, $h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus id$ from the server, the reader would calculate $id' = h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus (h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus id)$ and store *id'* in its memory, subsequency send N_{db}, $h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus id$ to the tag.
- After receiving N_{db}, $h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus id$ from the reader, the tag would calculate $h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus (h(id \oplus N_R \oplus N_T \oplus N_{db}) \oplus id)$, If the outcome is equal to *id* of the tag, then the object of mutual authentication is achieved, the tag should update $s = h(id \oplus N_R \oplus N_T \oplus N_{db})$, otherwise, the authentication is failed.

3.5.2 Result of verification

HLPSL specification of Jialiang et al.'s Protocol shown in appendix *B*. AVISPA tools detect trace of attack on tag authentication. Figure 3.6 shows the trace of attack on Wei et al.'s protocol.

We symbolize: n1(Nr) by N_R, Nr(5) by N_R', n5(Nt) by N_T, Nt(2) by N_T' and n2(Ndb) by Ndb. Several comments can be drawn from the trace:

- Msg1: The reader generates a nonce N_R and the intruder captures and stores the nonce in the course of the communication.
- Msg2: The intruder generates another instance of the nonce N_R' and sends it to the tag.

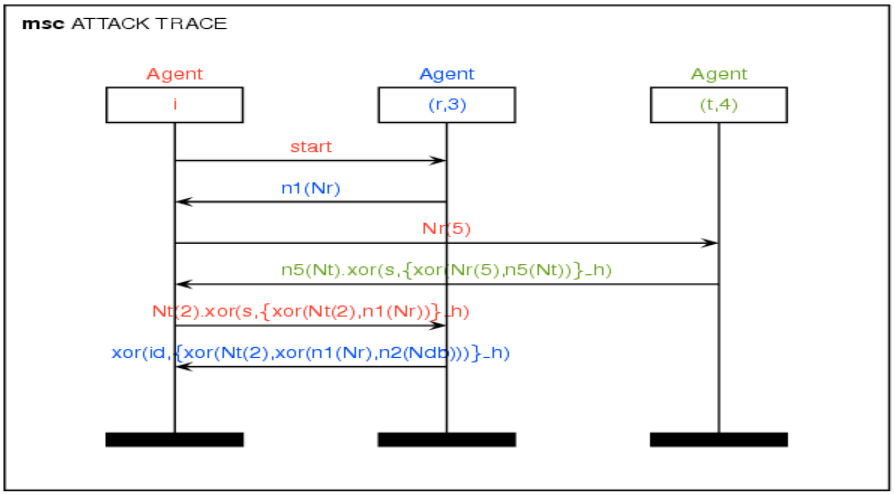

Figure 3.6: Trace attack on the Jialiang et al.'s protocol

- Msg3: The tag generates a nonce N_T and sends it with the xor of s and hash function $h(N_R' \oplus N_T)$ to the intruder.
- Msg4: The intruder generates a nonce N_T' and sends it with the xor of s and hash function $h(N_R \oplus N_T')$ to the reader.
- Msg5: The reader sends the function $h(N_R \oplus N_T \oplus N_{db}) \oplus N_{db}$ and N_{db} to the tag.
- The attack on tag authentication is realising in Msg4. We will describe the principle of this attack in the next section.

3.6 Algebraic Replay Attacks

In this section, we analyze the results of RFID authentication protocols and we cite the implementation and the countermeasure of ARA attacks.

Our results are based on the automatic verification of the authentication properties of each RFID authentication protocol. Concerning the message of tag authentication *Auth_tag*, the difference between these protocols is the type of one-way function (hash function and CRC) and the secret data which are shared between the tag and the reader (server).

For tag impersonation of the studies protocols, an intruder can store all the messages transmitted in a protocol run.

To tag impersonate, the intruder could replay $f(\alpha \oplus N_R \oplus N_T)$ if he ensures that $f(\alpha \oplus N_R \oplus N_T) = f(\alpha \oplus N_R' \oplus nt')$. The activate intruder can generate a new none and make an algebraic calculation of the type xor operation between numbers. Then, to satisfy this condition the intruder sets nt' to $N_R \oplus N_R' \oplus nt$. Here is the operation in detail:

$f(\alpha \oplus N_R \oplus N_T) =? f(\alpha \oplus N_R' \oplus \underline{N_T'})$

$f(\alpha \oplus N_R \oplus N_T) =? f(\alpha \oplus N_R' \oplus \underline{N_R \oplus N_R'} \oplus \underline{N_T})$ ➔ replace nt'

$f(\alpha \oplus N_R \oplus N_T) =? f(\alpha \oplus N_R' \oplus \underline{N_R'} \oplus \underline{N_R \oplus N_T})$ ➔ commutativity

$f(\alpha \oplus N_R \oplus N_T) =? f(\alpha \oplus 0 \oplus \underline{N_R \oplus N_T})$ ➔ nilpotence

$f(\alpha \oplus N_R \oplus N_T) = f(\alpha \oplus N_R \oplus N_T)$ ➔ neutral element

For tag impersonation in JDTL protocol, the principal vulnerability is the message of tag authentication $h(N_R \oplus N_T) \oplus s$. The intruder generates a nonce N_R' and sends in to the tag. The role of this nonce is obtaining the secret value *s*. The legitimate tag sends a message $h(N_R' \oplus N_T) \oplus s$ to the intruder. In this step, the intruder obtains the secret value *s*. Subsequently, the intruder generates a nonce Nt' of impersonation of tag and uses N_R of the legitimate reader to calculate $h(N_R \oplus Nt') \oplus s$. The intruder sends $h(N_R \oplus N_T') \oplus s$, N_T' to the tag. Then, impersonating the tag is successful.

All the studied protocols cannot resist attack of tag's authentication, and therefore an intruder can impersonate the tag. This type of attack is based on algebraic properties of algebraic operators (or, and, xor). The paper [DR09] aims to identify the algebraic problems which enable many attacks on RFID protocols. Toward this goal, three targeting types of attacks on RFID protocols have emerged authentication, untraceability, and secrecy are discussed.

The common theme in these attacks is the fact that the algebraic properties of operators (e.g. *xor* operator) employed by the protocols are abused. The methods used to find algebraic replay attacks are sufficiently straight-forward. The algebraic replay attacks in RFID authentication protocols are described in some works such as [DR08, CS09, CDP09, Mih11, JF12].

The relay attack system can use two transponders in order to relay the information that a reader and a token exchange during a cryptographic challenge response protocol. A

proxy-token device is placed near the real reader and a proxy-reader device is placed near the real token, possibly unknown to its holder. Information can therefore be forwarded over a great distance if a suitable communication medium is chosen between the proxy-token and proxy-reader. As a result, the reader will report that it has verified the presence of a remote token and thus provide access to the intruder [Han06].

Practically, the ARA system is based on relay attack system. The difference between this system and relay attack system is: this system supports Dolev-Yao attack model (see section 2). Therefore, the proxy system can generate a random number and compute xor operation between numbers. The process of attack system for Wei et al.'s protocol is as following (see figure 3.7):

- Legitimate reader generates a nonce N_R and sends it to the proxy-token.
- Proxy-token receives it and blocks it; the proxy-token generates a nonce N_R' and forwards this nonce to the proxy-reader through the fast communication channels.
- Proxy-reader fakes the real reader, and sends N_R' to the legitimate tag.
- Legitimate tag computes a new nonce N_T and computes hash function $h(s \oplus N_R' \oplus N_T)$ and transmits it to the proxy-reader.
- Proxy-reader receives it and calculates the new $N_T' = N_R \oplus N_R' \oplus N_T$ and forwards this message and the received hash function to the proxy-token through the fast communication channel.
- Proxy-token forwards N_T' and $h(s \oplus N_R' \oplus N_T)$ to the real reader.

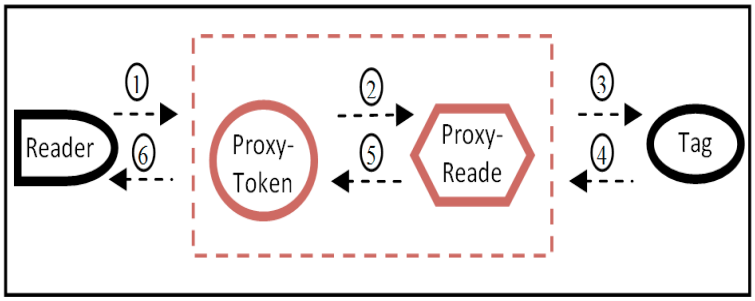

Figure 3.7: Attack System

3.7 Proposed Solution

Concerning the Wei et al.'s protocol, the proposed solution is to change the primitive XOR ($\oplus$) between the nonce N_R and N_T by the concatenation ($\|$). Therefore, the new hash

function is $h(N_R \| (N_T \oplus s))$. Automated verification of Wei et al.'s protocol after correction gives the following result:

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  UNTYPED_MODEL
PROTOCOL
  /home/avispa/web-interface-
computation/./tempdir/workfileCUQheG.if

GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 708 states
  Reachable  : 364 states
 Translation: 0.30 seconds
```

Figure 3.8: verification result of Wei et al.'s protocol after correction

This result showed clearly that there is no attack detected during authentication. We can thus conclude that this protocol is safe.

For tag impersonation in Jialing et al.'s protocol, the principal vulnerability is the message of tag authentication $h(N_R \oplus N_T) \oplus s$. The intruder generates a nonce N_R' and sends it to the tag. The role of this nonce is obtaining the secret value s. The legitimate tag sends a message $h(N_R' \oplus N_T) \oplus s$ to the intruder. In this step, the intruder obtains the secret value s. Subsequently, the intruder generates a nonce N_T' of impersonation of tag and uses N_R of the legitimate reader to calculate $h(N_R \oplus N_T') \oplus S$. The intruder sends $h(N_R \oplus N_T') \oplus s$, N_T' to the tag. Then impersonating of the tag is successful. We propose to use the message of tag authentication in WHC protocol corrected as $H(N_R \| (N_T \oplus S))$.

Therefore, the principal vulnerability in the studied protocols (presented in Table 3.1) is the use of xor operator in one-way function. Consequently, the solution is to change the primitive XOR ($\oplus$) between the values of one-way function (α, N_R, N_T) by the concatenation ($\|$). Therefore, the new one-way function is: $f((\alpha \oplus N_R) \| N_T)$or $f(\alpha \| (N_R \oplus N_T))$.

3.8 Conclusion

We have presented in this chapter different protocols using xor-operator and one-way functions. The one-way functions in the studied protocols are: hash function and CRC function. Our security analysis of these RFID authentication protocols by automatic formal tools. We showed that the verified protocols cannot resist RFID tag authentication attack therefore; an intruder can impersonate the tag.

The detected attack is the type of algebraic replay attacks (ARA) on tag authentication. The principal cause of the described attacks in our work is the abuse of the proprieties of xor-operator in the transmitted messages. The proposed solution for this attack is correcting the use of xor-operator and replacing it by concatenation operator.

Using the obtained results of this chapter, we propose a new protocol based on hash function and or-exclusive operator for combined RFID-Biometric system. The details will be presented in the next chapter.

Chapter 4

Hash-based Authentication Protocol for RFID-Biometric System

4.1 Introduction

The protocols of identity verification which allow access are called the authentication protocols. In RFID systems, several RFID authentication protocols have been developed (see chapter 2 and 3). The difference between the proposed protocols lies in the realized properties of security and the complexity of implementation. Most of these protocols use only tag RFID for access control. On the contrary systems with smartcards there are several authentication protocols based on the biometric technology.

We are interested in access control applications. Physical access control consists in verifying if a person asking to reach a zone (e.g. building, office, parking, laboratory, etc.), has the right necessities to make it. Technique of access controls are based on the following criteria: what one possesses (smartcard, tag RFID), what one is (biometry: fingerprints, face, iris, etc.), what one knows (e.g. password), or on a combination of these criteria.

In this chapter, we propose a hash-based authentication protocol for RFID-Biometric system (RBioA). Our protocol requires a PRNG, a robust hash function and Biometric hash function. The Biometric hash function is used to optimize and to protect biometric data. We prove the security properties of our proposed protocol by AVISPA tools. To estimate these performances, we will compare it with the other RFID protocols and the

biometric protocols of smart cards. The work of this chapter is based on our papers [CCB11, CCB12a].

The remaining part of this chapter is organised as follows. In section 4.2, we show different proposed biometric authentication protocols. Works which describe different implementations of combined systems biometric-RFID are summarized in section 4.3. In section 4.4, we describe components of our system. Section 4.5 details registration phase and mutual authentication of our RBioA protocol. Section 4.6 gives security analyses of RBioA protocol. Section 4.7 analyses of RBioA protocol in term of performance. Finally, this chapter terminates by a conclusion.

4.2 Biometric authentication Protocols

Biometry is widely used in the authentication protocols of smart cards applications [KZW08, LH10, LCC10]. The use of these protocols in RFID systems depends on the availability of computer resources (memory, complexity, performance, etc.), in the constituents of RFID systems and especially the RFID tag. The recent protocol [LCC10] requires the calculation of seven operations of the function h in the phases of login and authentication and requires $4l$ as storage space in the tag. This number of calculations and this storage space influences negatively on the efficiency of a RFID protocol. Another difficulty concerns "Matching" treatment. In the biometric authentication protocols of smart card, this part is made with the technique Match-on-card.

4.3 Implementation of RFID-Biometric system

Concerning the material implementation of combined systems biometric-RFID, we shall quote two recent works. Rodrigues and al. [RHV09] propose a decentralized authentication solution for embedded systems that combine both token-based and biometric-based mechanism authentication. Aboalsamh [Abo10] proposes a compact system that consists of a CMOS fingerprint sensor (FPC1011F1) is used with the FPC2020 power efficient fingerprint processor; which acts as a biometric sub-system with a direct interface to the sensor as well as to an external flash memory for storing finger print templates.

An RFID circuit is integrated with the sensor and fingerprint processor to create an electronic identification card (e-ID card). The e-ID card will pre-store the fingerprint of the authorized user. The RFID circuit is enabled to transmit data and allow access to the user, when the card is used and the fingerprint authentication is successful.

4.4 System model

The proposed system of authentication is based on the combination of two sub-systems: an RFID system and a biometric system.

4.4.1 RFID system

RFID system consists of: a tag *T*, a reader *R* and a server *S*.

- *Tag:* the tag stores the identity (id) and the biometric hash function of the template of the person (*GB*). This *id* is strictly confidential and is shared between the database of the back-end server *S* and the tag *T*. The tag can generate random numbers, and calculation of the hash function *h* of a number. Standard ISO and EPC GEN2 (Generation 2) support the producing of the random numbers (nonces) in the tag.
- *Reader:* The reader *R* can generate also the random numbers. The communication between the reader and the server is secured,
- *Server:* the server has two main functionalities: (1) for the biometric system: extraction of the characteristics of a biometric modality to create a model or template *B*. (2) For the RFID system: it contains the database which includes the list of the identity of tags id.

4.4.2 Biometric system

The biometric system consists of two entities, a sensor (*SR*) and a server (*S*). The biometric device in our system is *Sensor,* this biometric sensor is an electronic device used to capture a biometric modality of a person (fingerprint, face, voice, etc.).

Biometric data can be stored in the tag or in the database. The biometric template will be stored in the tag. It offers a greater privacy and the mobility for the user. This assures also that information will always be with the user's tag.

Storing the raw biometric data typically requires more substantial memory. For example, a complete fingerprint image will require 50 to 100 Kbytes, while a fingerprint template requires only 300 bytes to 2 Kbytes [Sma11]. This condition is not always sufficient especially for the type of passive RFID tags. In our system, a practicable solution to optimize and to protect biometric data is the hash function. This function of template allows pressing the biometric template into an acceptable size.

The problem which lies with the hash functions standard (e.g. SHA-1 , MD5 , SHA-256, …) is the comparison between two templates: the template which is protected in the tag a *h(B)* and the template which is generated from the capture *h(B')*. Equality *h(B) = h(B')* for the same person is not always assured, because *B'* is a dynamic template where the person never keeps the same biometric features, (e.g. movement of the finger during the purchase), which implies the existence of a rate of error. We will cite two research works:

Sutcu and *al.* [SSM05] propose a secure biometric based authentication scheme which fundamentally relies on the use of a robust hash function. The robust hash function is a one-way transformation tailored specifically for each user and based on their biometrics. The function is designed as a sum of properly weighted and shifted Gaussian functions to ensure the security and privacy of biometric data. They also provide test results obtained by applying the proposed scheme to ORL face database and designating the biometrics as singular values of face images.

A. Nagar and al. [NNJ10] propose six different measures to evaluate the security strength of template transformation schemes. Based on these measures, they analyze the security of two well-known template transformation techniques, namely, Biohashing and cancelable fingerprint templates based on the proposed metrics.

4.5 Description of our RBioA protocol

The proposed Protocol RBioA is divided into two processes: the phase of registration and the phase of mutual authentication. Steps detailed by two processes are described below.

4.5.1 Registration Phase

This registration phase is also called setup phase. The objective is to create a biometric template and store it in the related declared identity (see the figure 4.1). In this phase, it has to apply the following steps to obtain the RFID tag.

Step 1: the authorized user inputs his/here personal biometrics, to pass it on to the server of the trusted registration center *RC*.

Step 2: the *RC*, after extraction of biometric characteristics, creates a biometric template *B*, and computes the biometric hash function *GB* such as $GB = g(B)$.

Step 3: Then, the registration center stores the information $\{id, GB\}$ in the user's tag and sends it to the tag through a secure channel.

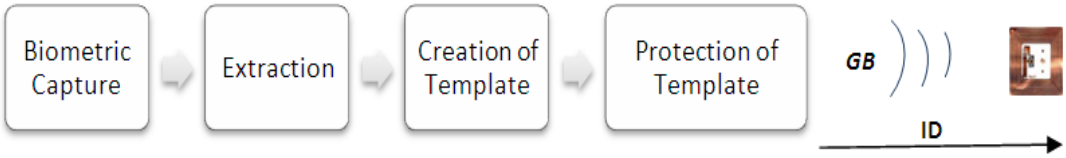

Figure 4.1: Registration Process of RBioA protocol

4.5.2 Mutual Authentication Phase

According to the order of the passed on messages, the phase of mutual authentication is described as below (to see Figure 4.2):

Step 1: Tag authentication

Step 1.1: R generates random nonce Nr and sends it as a query command to T.

Step 1.2: the tag found in the step 1 generates a nonce Nt and computes P such as: $P=LH(id \oplus Nt \| Nr)$

Step 1.3: the tag sends P with the nonce Nt to the reader RFID,

Step 1.4: the reader resends the successful P message, Nt and the nonce Nr to the server.

Step 1.5: from the database, the server looks for a certain idi (such as $1 \leq i \leq n$, n is the number of tags) to compute $Pi=LH((idi \oplus Nt) \| Nr)$, and make the following comparison:

$$Pi \ ?= \ P$$

If it is found, the tag crosses the authentication of the tag and is considered as legitimate, otherwise it is set to end.

Step 2: Reader authentication

Step 2.1: the server computes and sends Q to the reader;

$Q = RH(id_i \oplus Nt \| Nr)$ such as $id_i = id$

Step 2.2: the reader sends the Q message in the tag.

Step 2.3: the tag computes $RH_R(id \oplus Nt \| Nr)$ and verifies if:

$$Q \ ?= \ RH((id \oplus Nt) \| Nr)$$

If they are equal, the authentication of the reader is successful; otherwise the authentication of the reader has failed.

Step 3: Biometry Verification

Step 3.1: the tag computes $M1 = h(id \| Nt \| Nr)$ and makes operation or-exclusive of M1 with GB and Nt. The resultant message is $M = M1 \oplus GB \oplus Nt$.

Step 3.2: the tag sends M to the reader RFID, and the reader resends received message to the server.

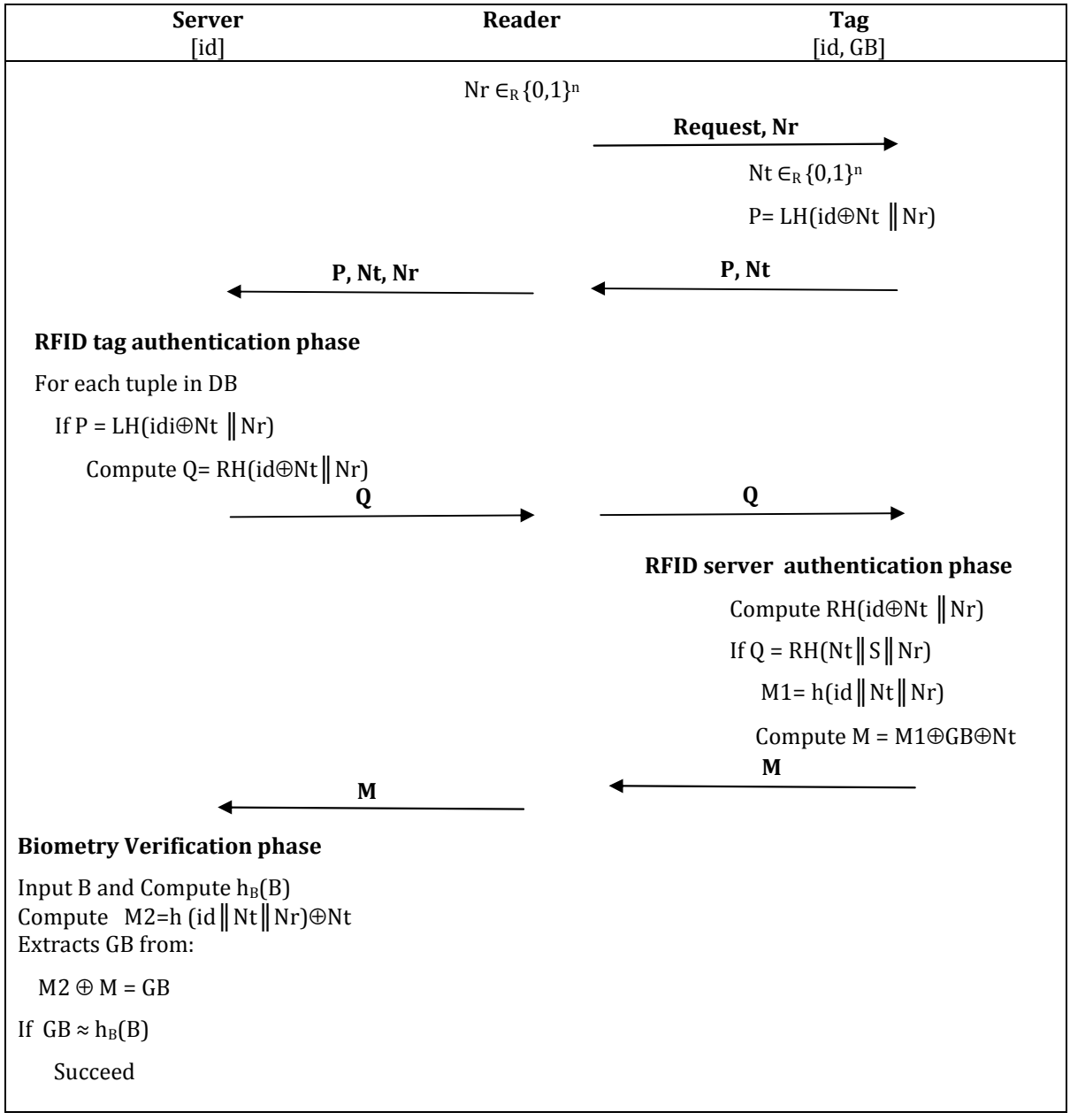

Figure 4.2: Authentication phase of proposed RBioA Protocol

Step 3.3: after acquiring the biometry of the user from the sensor, it sends it to the server. The server extracts biometric characteristics and generates the template *B*. the server computes the biometric hash function of the

template $h_B(B)$.

Step 3.4: from the database, the server computes `M2=h (idi‖Nt‖Nr)⊕Nt`, such as `idi= id` (of the *step 1.5*), and extracts GB from:

`M2 ⊕ M = GB`

Step 3.5: to make the comparison of type 1:1 of $h_B(B) \approx GB$, if it is confirmed, the person is a trusted user, otherwise, the bearer of the tag is illegitimate, the information of failure will be sent to the reader, the protocol is interrupted.

4.6 Security Analysis

4.6.1 Automated Verification

To verify the security of our proposed protocol, we specify it by HLPSL. Then, we verify it by AVISPA tools. HLPSL specification of our proposed protocol is shown in appendix C.

The verified properties are: secrecy of the identity id (sec_id_TR and sec_id_RT respectively), the secrecy of the template B (sec_b), tag authentication (aut_tag) and reader authentication (aut_reader). These properties are specified in HLPSL as follows:

```
goal
    secrecy_of sec_b, sec_id_TR, sec_id_RT
    authentication_on aut_reader
    authentication_on aut_tag
end goal
```

As for the authentication, there are two possible attacks: the replay attack and the man-in-the-middle attack. For this, we uses two types of specification in the role *environment*.

a) Replay Attack

In the replay attack, the intruder can listen to the message of answer of the tag and to the reader. It will broadcast the message listened without modification to the reader later.

Specification below of the role *environment* in HLPSL depends on the treatment of two identical sessions between the same tag and the same reader (`t` and `r`). This scenario allows discovering the potential existence of attacks of the type replay attack.

```
role environment() def=
const t,r : agent,
      id,b : text,
```

```
      h,g,left,right : hash_func
intruder_knowledge = {t,r,h,g,hright,hleft}
composition
session(t,r,id,b,h,g,hright,hleft)/\
session(t,r,id,b,h,g,hright,hleft)
end role
```

After the verification of this protocol by AVISPA tools, result is showed in Figure 4.3. This result means in clear that there is no replay attack. We can thus deduct that the diagnosis of AVISPA&SPAN tools for this protocol is secure.

```
      SUMMARY
        SAFE
      DETAILS
        BOUNDED_NUMBER_OF_SESSIONS
        UNTYPED_MODEL
      PROTOCOL

C:\progra~1\SPAN\testsuite\results\BioMRFID.if
      GOAL
        As Specified
      BACKEND
        CL-AtSe
      STATISTICS
        Analysed   : 600 states
        Reachable  : 188 states
        Translation: 0.01 seconds
       Computation: 0.02 seconds
```

Figure 4.3: verification result of RBioA Protocol

b) Main-in-the-middle Attack

The scenario of the role *environment* below allows discovering the the potential existence of attacks of this type.

```
role environment() def=
const t,r : agent,
      id,b,idti,idri,bti,bri : text,
      h,g,hright,hleft : hash_func
intruder_knowledge={t,r,h,g,hright,hleft, idti,idri,
                    bti, bri}
composition
```

```
        session(t,r,id,b,h,g,hright,hleft)
  /\    session(t,i,idti,bti,h,g,hright,hleft)
  /\    session(i,r,idri,bri,h,g,hright,hleft)
end role
```

The result of the check with this scenario is the same as with the scenario a). We can thus deduct that this protocol is resistant to the attack of the "man in the middle".

4.6.2 Formal Analysis

Using Ouafi-Phan model [OP08], we verify untraceability property. During every session of authentication, an opponent can observe only the values of (Nt, Nr, M1, P, Q), where, Nt and Nr are random numbers and M1 and Q messages are the calculated right/ left part of the function $H((id \oplus Nt) \| Nr)$. The message P = $H(id \| Nt \| Nr) \oplus GB \oplus Nt$. The adversary cannot deduce the value of *id* because function $H(id \| Nt \| Nr)$ is very effective as is shown in the paper of [JW07]. In messages *M1, P* and *Q*, the adversary cannot correlate *id* and *B* because these two values are secret and Nt and Nr are random numbers changed in every authentication. So, an adversary cannot track tags.

4.6.3 Security Analysis

We now analyze the security properties of the proposed protocol as follows: desynchronization resilience and Denial of service (DOS) attack prevention. In the Table 4.1 below, a comparison of the security with protocols mentioned earlier is given [WSRE03, LHLL05, CH07, LHYC08].

RFID Protocol *(static ID)*	[WSRE03]	[LHLL05]	[CH07]	[LHYC08]	**Our Protocol**
Mutual Authentication	+	+	+	+	**+**
Replay attack prevention	-	+	-	+	**+**
Untraceability	-	+	+	+	**+**
DoS attack prevention	-	-	+	+	**+**
Desynchronization resilience	+	+	+	+	**+**

Table 4.1: Security comparaison of RBioA protocol

c) Desynchronization Resilience

Our protocol belongs to the static mechanism *id* where the identifier of the tag is fixed. So, in the case of the loss of message, failing of energy or the loss of connection

with the server during the authentication, this will not affect the database of the server and will not become an obstacle to the protocol.

d) DOS attack Prevention

There are several categories of Dos attacks, one is to desynchronize the internal states of two entities, and the other is to exhaust the resources of the parties involved. For RFID authentication protocols, researchers are concerned about desynchronization.

As for our protocol, the internal state id is kept static and not changed during the authentication process. So, it can resist the attack of denial of service.

4.7 Performance analysis

Table 4.2 illustrates the storage cost, the communication cost, and the computation cost of entities. The computation cost is a function of the number of operations of the hash function in login's phase and the authentication on the smartcard for the biometric protocols, as well as of the number of operations of the hash function on the tag in RFID protocols.

Computation Cost The tag used in the protocol proposed by Lee et al. [LHYC08] and the smart cards of the biometric protocols require an important number of operations for the hash function. On the contrary, in the protocol of Chien and Huang [CH07], it requires a random numbers generator with an input number, but it is necessary not to forget the replay algebraic attack. In our protocol, we require two operations of calculation of function h in the tag, so these calculations are effective for RFID tags.

Communication Cost Communication cost between a tag and a reader consists of: the number of message exchanges, and the total bit size of the transmitted messages per each communication. Concerning our protocol, the total of the bits of the messages of communication tag to the reader is: 2½l and for the message of communication reader to tag it is: ½l. Compared to the other protocols of smart cards, the performance of the communication of our protocol is more effective.

Storage Cost The amount of storage needed on the back-end server is also another important issue. In the biometric protocols [KZW08, LH10], the smart card requires 3l bit and $4l$ for the protocol [LCC10]. In our protocol, the tag requires 2l bit to store the identity (id) and the function h of template (GB).

Protocol		Computational Cost Tag/Smartcard	Storage Cost	Communication Cost		
				R →T	T →R	∑
RFID	[CH07]	*1g*	*2l*	*½l*	*1½l*	*2l*
	[WSRE03]	*1h*	*1l*	-	*2l*	*2l*
	[LHLL05]	*3h*	*1l*	*3l*	*3l*	*6l*
	[LHYC08]	*4h*	*2l*	*1l*	*2l*	*3l*
Smartcard	[KZW08]	*4h*	*3l*	*2l*	*3l*	*5l*
	[LH10]	*4h*	*3l*	*2l*	*3l*	*5l*
	[LCC10]	*3h*	*4l*	*2l*	*3l*	*5l*
Our RBioA protocol		***2h***	***2l***	***½l***	***2½l***	***3l***

Notations: *h* - the cost of a hash function operation,

g - random number generator with an input number,

l: size of required memory.

Table 4.2: Performance Analysis of RBioA protocol

Consequently, in the implemented protocols, the tag requires only 2*l* bits at most of the memory, which is adapted to tags with weak cost.

We can conclude that our protocol is effective and adapted to RFID tags as far as the computation cost; the storage cost and the communication cost are concerned.

4.8 Conclusion

RFID systems can be applied in various areas, among the important ones of them, the access control. This work proposed a new RFID authentication protocol (RBioA). For an authentication phase, RBioA protocol is based on the combination of RFID tag and biometric data. Our proposed protocol realizes the secrecy private data, the tag authentication and the reader authentication. Experimental tests (with AVISPA and SPAN tools) proved its efficiency. The careful security analysis showed that the new protocol can resist man-in-the-middle attack, replay attack and the tracing attack. Moreover, the performance evaluation showed that the new protocol is compatible with the constrained computational and memory resources of the RFID tags.

Our RBioA protocol is of category hash-based RFID protocols that need exhaustive research to obtain the value tag's identifier, i.e. complexity is $O(n)$. In the next chapter, we will show a review of code-based RFID protocols which don't need an exhaustive research i.e. the complexity is $O(1)$.

Chapter 5

RFID Authentication Protocols based on Error-Correcting Codes

5.1 Introduction

In the literature on design of RFID authentication protocols, we can find several categories according to various primitives requirement (described in chapter 2). The code-based cryptography is a very important research area and it is applied in different schemes. The major problem was the size of public key; recently, code-based cryptosystems were presented with small key sizes. In the majority of RFID authentication protocols, the tag does not require a generator matrix or other matrices, but it stores the codeword with the necessary information.

In this chapter, we review various and recent RFID authentication protocols based on error correcting codes. These protocols use various schemes based on coding theory: randomized Niederreiter cryptosystem [SKI06, CKMI07], error-correcting code with secret parameters [Par04, Chi06, CL09], Quasi-Dyadic Fix Domain Shrinking [SKI10], randomized McEliece cryptosystem [MM12], combination between number theory [Chi13], and based on Quasi Cyclic-Moderate Density Parity Check (QC-MDPC) McEliece cryptosystem [LYL14].

Among these protocols, and in our paper [CCCB15a], we provide enough evidence to prove that two recent RFID authentication protocols are not secure. These protocols are: Malek and Miri [MM12], and Li et al. [LYL14].

5.2 Park's Protocol

5.2.1 Review of Park's protocol

Park [Par04] proposed a one-way authentication protocol to provide untraceability, it is based on the secret-key certificate and the algebraic structure of the error-correcting code. We note that this protocol is designed for wireless mobile communication systems. We study this protocol because the computational capabilities of mobile subscriber is limited like the RFID tag, we denote mobiles subscriber (MS) as tag *T* and the authentication server (AS) as reader *R*.

a) Initialization phase:

The *T* computes and stores the following tokens x_i with $x_{i-1}=g_0(x_i)$, for $i=s,s-1,...,1$. The *T* sends the root authentication token x_0 to the *R*. The *R* computes a symmetric-key certificate of the tag $SC = \{id||x_0\}\_k_R$, where the secret key k_R is only known by R. Then, the encoding of SC with matrix *G*, the encoded certificate is $c=SC.G$. Finally, the encoded certificate *c* is sent to the *T* in a secure channel.

b) The authentication phase

The authentication phase is depicted as follows (see Figure 5.1):

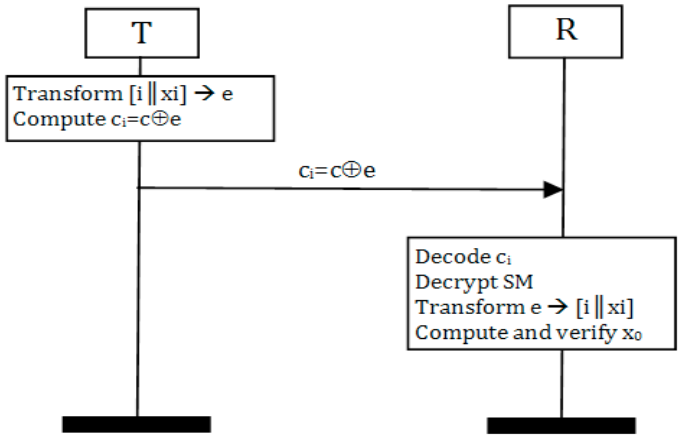

Figure 5.1: Park's Protocol

- From application $\emptyset_{n,t}([i \parallel x_i])$ of enumerative method, *T* generates the error vector *e* of session (*i*) ,and computes $c_i=c\oplus e$ and send it to R.
- The *R* decodes the received word $c\oplus e$ using the corresponding decoding algorithm, obtaining (*e*,*SC*).

- *R* decrypts *SC* to obtain the identifier *id* and x_0 to verify the token x_i in the error vector.
- *R* applied $\emptyset_{n,t}{}^{-1}(e^{(i)})$ to obtain $[i||x_i]$, and computes a series of authentication tokens $g_0(x_i)$, $g_0(x_{i-1})$,…, $g_0(x_1)$,, and verify if $g_0(x_1)$ is the same as the value x_0 retrieved from the secret-key certificate, if equal, then tag authentication is successful.

5.2.2 Traceability Attack

Figure 5.2 shows the message transmission of the traceability attack, and the following is the detailed description of each step:

- At session (*i*), the intruder intercepts $c \oplus e^{(i)}$,
- At session (*j*), it intercepts $\oplus e^{(j)}$.

The Hamming weight of $(c \oplus e^{(i)}) \oplus (c \oplus e^{(j)})$ is less than 2*t*, and the codeword *c* fixed for all sessions leads to attack on message-resend attack , and implicates an attack on untraceability and on confidentiality of *c* (see Figure 5.2). This attack is described also by [Dom06].

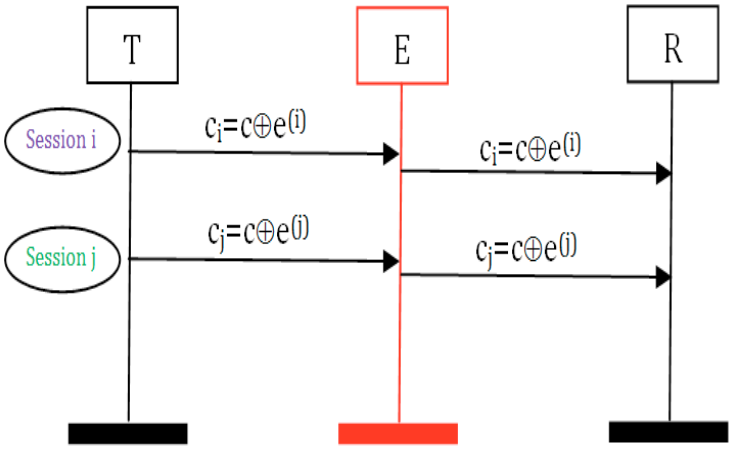

Figure 5.2: Traceability attack on Park's Protocol

5.2.3 Desynchronization attack

Other vulnerability of the Park's protocol is of type deynchronization attack. Figure 5.3 shows the message transmission of this attack, and the following is the detailed description of each step:

At session (i), the intruder blocks the message c_i. In new run of protocol, the value of session *i* stored in the reader is different from *i* stored in tag this implicates that the tag and the reader are not correlated and will be in a desynchronization state.

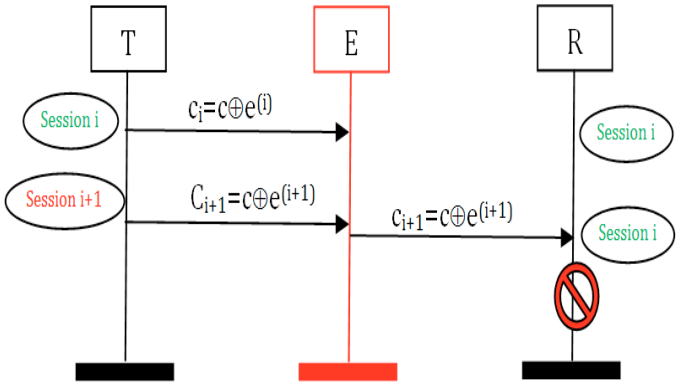

Figure 5.3: Desynchronization attack on Park's Protocol

5.2.4 Performance analysis

Concerning storage cost, if the number of authorized sessions *s* is very large, the tag needs important storage space for stocking all the values of $g_0(x_i)$. For example, if s=1000 times and the length of $g_0(.)$ is 100 bits, then the tag requires 97.66 Kb for all tokens x_i.

5.3 Chien's Protocol (2006)

5.3.1 Review of Chien's Protocol

In paper [Chi06], the author proposed two authentication protocols for RFID systems oriented to access control applications. Firstly protocol is based on hash function, the second one is based on error-correcting codes. We are interest by this last protocol.

a) Initialization phase

The server (S) generates a unique key for each tag, $key=h(K_{svr} \| id)$, where K_{svr} is the master key of the server. The server also selects a random seed p_0 and computes $p_{n'}=h^{n'}(p_0)$ and the secret certificate for the tag as $Cert_{tag} = E_{K_{svr}}(id \| p_n \| key \| valid_{time})$ where $valid_{time}$ denotes the valid time period of this certificate, $h^{n'}(.)$ denotes hashing *n'* times and *n'* denotes the maximum allowed authentications per tag for each imprinting.

After the certificate becomes expired, the tag should be re-imprinted. The S further encodes the certificate as a codeword $c_{tag}=Cert_{tag}.G_{server}$. The tag stored the values (key, c_{tag}, p_0).

b) Authentication phase

The authentication phase is depicted as bellow (Figure 5.2):

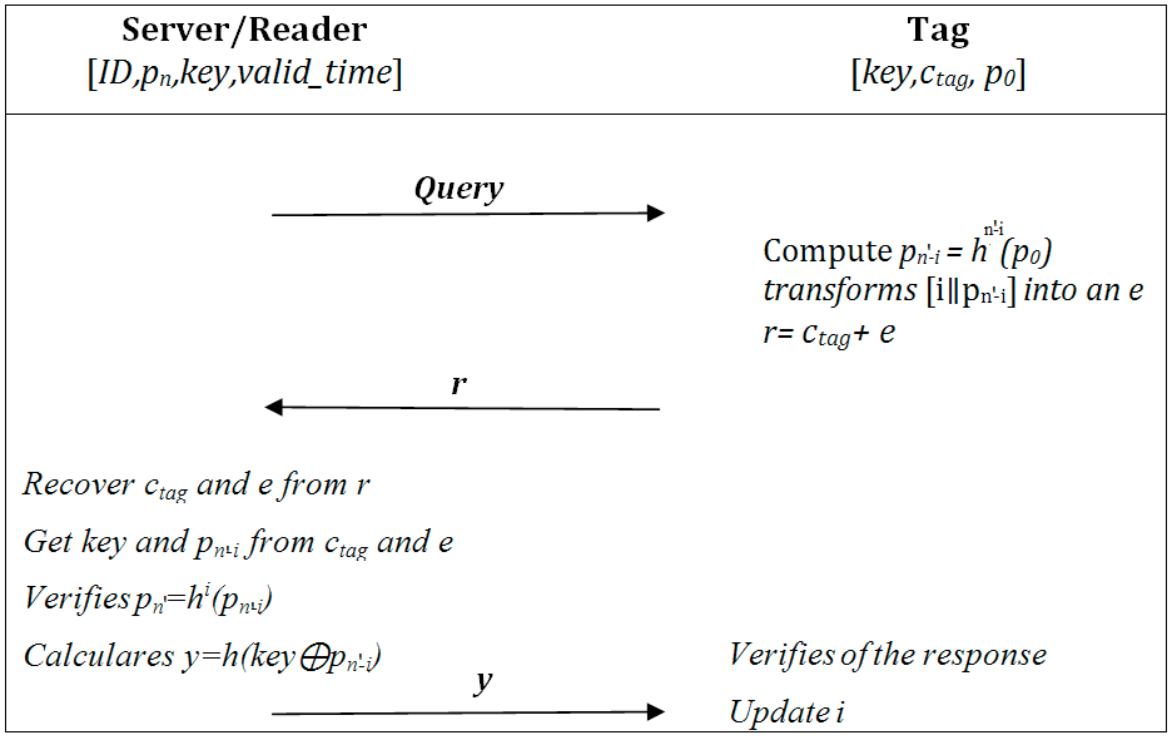

Figure 5.4: Chien's Protocol (2006)

- The *R* sends the query message to the *T*.
- The T computes $p_{n'-i}=h^{n'-i}(p_0)$, and transforms $[i \| p_{n-i}]$ into an *e* using Algorithm of tranform string bits to error vector (see chapter 2). It adds *e* and c_{tag} to get the transmission vector $r=c_{tag}+e$. The *R* forwards the received *r* to the *S*.
- *S* uses H_{server} and Algorithm of transform error vector to string bits to derive the value $[i \| p_{n'-i}]$ and c_{tag}. It decrypts c_{tag} to get the data $(id \| p_n \| key \| valid_{time})$. The server verifies the tag by checking whether the equation $p_n?=h^i(p_{n'-i})$ holds. If so, the *T* is authenticated, and the *S* updates the local value *i* and goes to the next step; otherwise, it responds an error message to the *R*.
- *S* computes and sends back "success" and $h(key \oplus p_{n-1})$ to the reader. The reader forwards this value to *T*, and *T* can verify the validity of the response and then updates its local value *i*.

5.3.2 Desynchronization attack

In each run of protocol, the tag and the reader store the number *i* of the last session. If the intruder blocks the last message from *R* to *T*, then, the value *i* of the session which is

stored in *R* is different than *T*, which implicates that *T* and *R* are not correlated. Then, the Chien's protocol (2006) does not resist desynchronization attack.

5.3.3 Performance analysis

Concerning the computational evaluation, in each run of protocol, the tag compute the hash function of p_0 (*n'-i*) times. For example If the number of authorized authentication *n'* is 1000, then $h(p_0)$ is computed 999 times for session i=1. Thus, this is an important computation and not compatible with low-cost tags.

5.4 Cui et al. Protocol

5.4.1 Review of Cui et al.'s Protocol

In paper [CKMI07], the authors proposed an authentication protocol based on randomized Niederreiter cryptosystem and amelioration of the protocol [SKI06].

a) ***Initialisation phase***

The identity of tag is uniquely mapped to an element *id*, *R* computes $c_2 = idH_2$ and sends it to *T* with matrix H_2.

b) ***Authentication phase***

The authentication phase is depicted as follows (see figure 5.5):

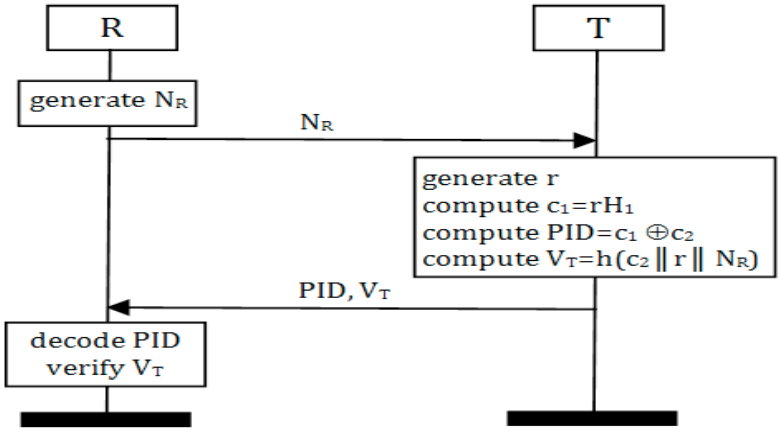

Figure 5.5: Cui et al.'s Protocol

- The reader *R* generates random number N_R and sends it to *T*.
- *T* generates random number *r* with length n_1 and weight t_1 and computes $c_1 = rH_1$ and $PID = c_1 \oplus c_2$.
- It computes $V_T=h(c_2 \| r \| N_R)$ and send *PID* with V_T to *R*.

- *R* decodes *PID* with private key to obtain *id* and *r*. Then hashes value $h(idH_2id\|r\|N_R)$ is compared with V_T, tag authentication is successful if they are equal, otherwise authentication has failed.

5.4.2 Security Analysis

In this protocol, the intruder could derive c_2 and matrix H_1 from a compromised tag. These data stored inside the legitimate tag are constant during its life. Therefore, this protocol does not achieve the forward secrecy. However, Cui et al's protocol does not achieve reader's authentication, consequently this protocol is one-way authentication.

5.4.3 Performance Analysis

As for the performance, the tag stores the public-key matrix H_1 to encrypt *r*. This is disadvantageous on two faces: requirement of space of non-volatile memory, and computation of ciphertext rH_1. The proposed solution is replacing this matrix by vector with length *n* bits using principle of quasi-cyclic codes. We use shifting circular in vector to calculate rows of public-key matrix H_1.

5.5 Chien and Laih's Protocol

5.5.1 Review of Chien and Laih's Protocol

Chien & Laih [CL09] proposed a lightweight RFID authentication protocol based on error-correcting codes. This protocol uses confusion scheme to avoid message-resend attack and related-message attack.

a) ***Initialisation phase***

Initially, *R* chooses randomly a secret linear code $C(n,k,d)$, as specified by its generator matrix *G*, and assigns row vectors $G[j]s'$ to *T* for $j=(z-1)*s'+1,\ldots,z*s'$, when *z* is order of tag. *R* maintains the information of each tag in its database *id*, *K* and indices of the assigned rows of *G*. Tag's memory stored *id*, *K*, vectors $G[j]s'$.

b) ***Authentication phase***

The authentication phase of the protocol is described as follows (figure 5.6):

- *R* generates a nonce N_R , and sends it with a query message to the *T*.
- *T* generates a non-zero codeword *c* via a random linear combination of row vectors $\{G[j]j=(z-1)*s'+1,\ldots,z*s'\}$ and randomly computes error vector *e*.

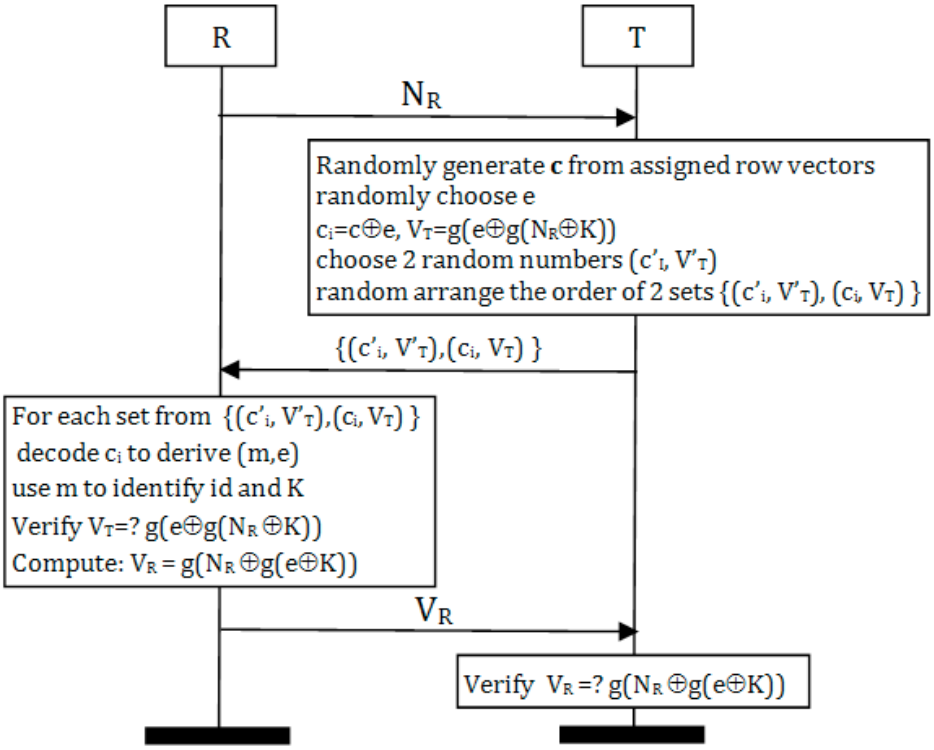

Figure 5.6: Chien and Laih's Protocol [CL09]

- It calculates c_i=c⊕e, V_T=g(e⊕g(N_R⊕K)) and generates two random numbers (c_i', V_T'), where $|c_i'| = |c_i|$ and $|V_T| = |V_T'|$.
- T sends the two sets $\{(c_i', V_T'), (c_i, V_T)\}$, to R.
- R uses the decoding algorithm to derive (m,e), where m can be used to identify the tag T and K verifies the equality {V_T?=g(e⊕g(N_R⊕K))} to accept tag's authentication.
- R computes V_R=g(N_R⊕g(e⊕K)) and sends it to T.
- T verifiesV_R?=g(N_R⊕g(e⊕K)). If they are equal, the reader's authentications successful; otherwise the reader's authentication has failed.

5.5.2 Security analysis

The tag stores *id*, *K* and row vectors *G*[*j*]*s*' are static information, therefore, this protocol does not achieve the forward secrecy. However, this protocol cannot prevent the tracing attacks [CCZ+14]. Authors used confusion scheme for avoid message-resend attack and message-related attack, but we can protect our protocol and reduce the communication cost and minimize computational operations by treating the weight of error vector.

5.5.3 Performance Analysis

As for the storage cost, if *s*' number of row vectors assigned to tag is important, then the space memory requires length of generator matrix *G* multiplied by total number of rows on tag and 2×|*K*|.

5.6 Sekino et al. Protocol

Sekino et al. [SKI10] proposed a challenge response authentication protocol based on Quasi-Dyadic Fix Domain Shrinking that combines Niederreiter personalized public key cryptosystem (P^2KC) [KI06] with Quasi-dyadic (Goppa) codes [MB09]. The principal objective of this approach is the reduction of size key matrix H_1 witch is stored in tag.

5.6.1 Review of Sekino et al. Protocol

a.) ***Initialisation phase***

The decryptor can learn who has generated the ciphertext, and also the system provides reduction of the encryption key size and reduction of the encryption computing. (P^2KC) can generate encryption key *ppk* from public key of Niederreiter PKC with *n* dimension vector *pv* (personalized vector). The sender encrypts plaintext by using *ppk* , with *ppk*(*H*1*, c*2*, t, Sub*), becomes (*n*−*k*)x*n*1 binary matrix H_1, dimension vector c_2 ans *Sub* is (n_1 + 2) sequence. The ciphertext is $c = (H_1^t)m \oplus c_2)$, where *m* is vector of length n_1. Decryption of (P^2KC) uses the decoding algorithm of Niederreiter PKC.

The public key *H* is produced with the structure of FQD (Flexible Quasi-Dyadic) and makes FDS (Fix Domain Shrinking) adjust to *H*.

b.) ***Authentication phase***

The authentication phase of this protocol is the same authentication phase of Cui et al.' protocol. The only difference between this protocol and Cui et al's protocol is articulated on the method of generation of public matrix. The method used to generate a parity-chek matrix of *t*×*n* is called Flexible Quasi-Dyadic. On the contrary, in the Cui et al.'s protocol, it requires a public matrix of *n*−*k*×*n*, where *t*<*n*−*k*.

5.6.2 Security Analysis

The information stored in tag c_2 and H_1 are static, therefore, this protocol does not achieve the forward secrecy.

5.6.3 Performance Analysis

Concerning storage space in tag, the authors reduce the space requirement from $(n-k)\times n_1$ in Cui et al' protocol into $(n-k)\times(n_1-(n-k))/t$, but it remains relatively for important the resources of low-cost tag.

5.7 Malek & Miri Protocol

5.7.1 Review of Malek and Miri Protocol

Malek and Miri [MM12] proposed a RFID authentication protocol based on randomized McEliece public-key cryptosystem. In this protocol, the tag can communicates with a set of authorized tags. So, it is possible to have different parameters for different readers to be stored in the memory of tag.

a) ***Initialisation phase***

In the initialization phase, the trusted center (e.g. server) selects a binary string *id*. Then it generates a random string *r* that uniquely identifies the tag with *id*. The trusted center encrypts [$r \parallel id$] using the randomized McEliece cryptosystem. The trusted center outputs $rG1 \oplus idG2$. Then it stores $\{rG1 \oplus idG2, id\}$ in the tag's memory. The data stored in the reader are private matrices and a database composed of *{idR, r, id}*, where *idR* is the reader's identifier. We note that in this protocol, the tag can communicate with a set of authorized readers. So, it is possible to have different parameters for different readers to be stored in the tag's memory.

b) ***Authentication phase***

The authentication phase of the protocol is described as follows (figure 5.7):

- reader *R* sends the query message with id_R to the *T*.
- *T* searches its memory to find the values *id* corresponding to id_R. If *T* finds the corresponding values, it generates a random error vector *e*.
- *T* computes $y=rG_1 \oplus idG_2 \oplus e$ and sends it to *R*.
- *R* decrypts *y* to retrieve (*r*,*id*) and *e* and verifies the received values with *id*,*r* stored in the database. If tag's authentication is successful, *R* generates a new random vector *p* with length *n* and computes a circular matrix A_p from *p*. It

sends the response set $\{d_0,d_1\}$ to T, where $d_0=rG_1 \oplus idG_2 \oplus p$ and $d_1=id \oplus h(eA_p)$, the length of output of hash function $h(.)$ is k_2.

- T computes $d_0 \oplus rG_1 \oplus idG_2$ to find p and uses its value to generate an A_p. It then, verifies the equality of $d_1 \oplus h(eA_p)$ and id. When the reader's authentication is successful, the tag requests OK to R.
- R generates a new random r' and computes $y'=r'G_1 \oplus idG_2 \oplus e$. It sends it to T.
- T refreshes its memory content by replacing $\{r'G_1 \oplus idG_2, id\}$ with $\{y' \oplus e, id\}$ and terminates this session.

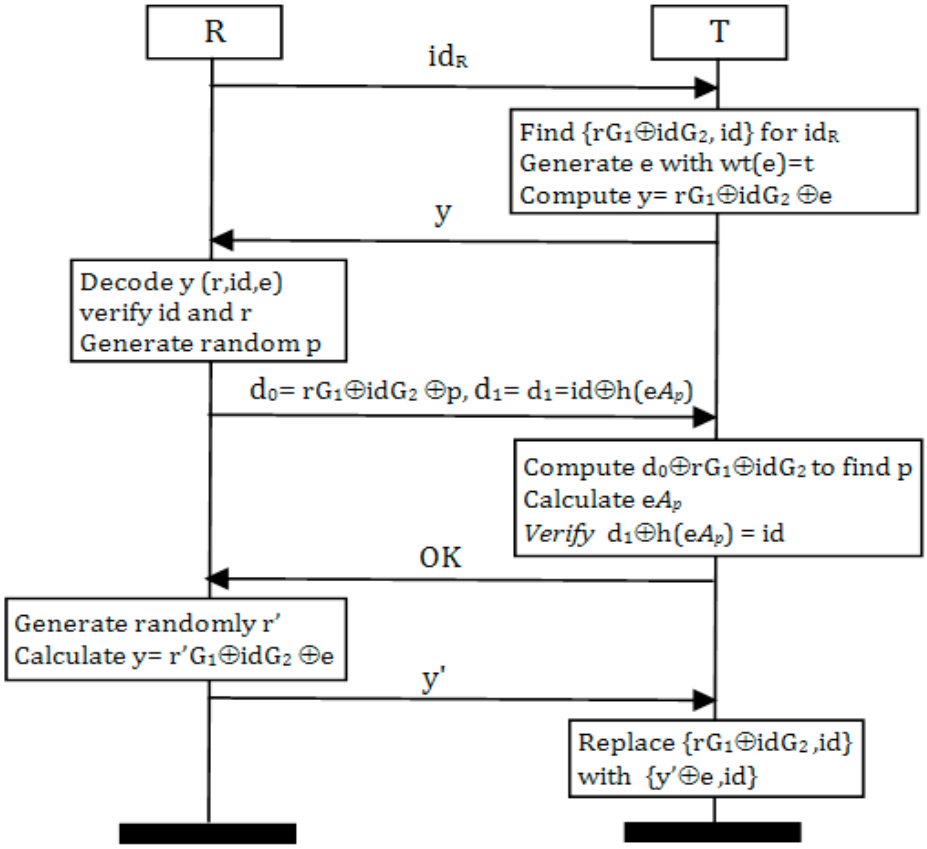

Figure 5.7: Malek and Miri's Protocol [MM12]

5.7.2 Desynchronization attack

We assume that the adversary has a complete control over the channel of communication between the reader and the tag. It can intercept any message passing through the network, modify or block messages, and it can also create new messages from its initial knowledge.

Figure 5.8 shows the message transmission of the desynchronization attack, and the following is a detailed description of each step:

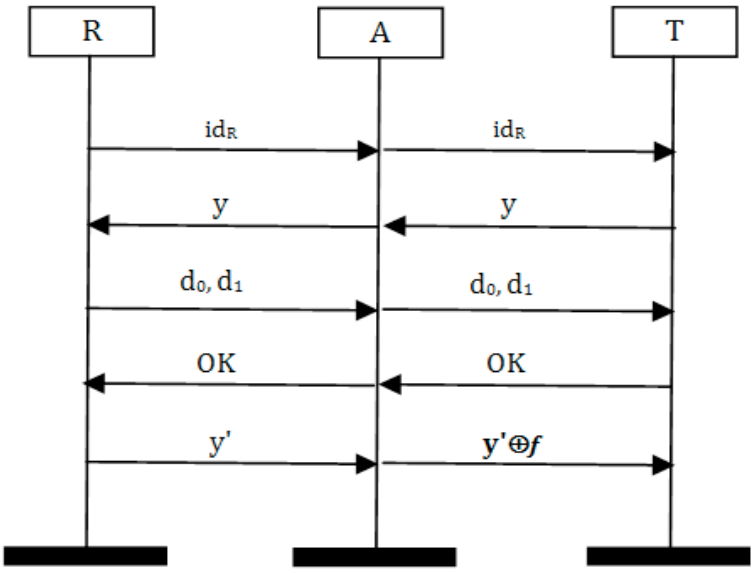

Figure 5.8: Desynchronisation attack on Malek and Miri's protocol

- At session (i), we suppose that the system is processing normally, steps of the tag's authentication and the reader's authentication are successful. T requests OK to R and the adversary intercepts the messages transmitted between R and T.
- R generates a new random r', computes $y' = r'G_1 \oplus idG_2 \oplus e_i$, and sends it. R updates the value of r by r'.
- The intruder blocks the message y', generates a vector *f*, and computes y'⊕ *f*. It sends it to T.
- T updates the stored data $\{rG_1 \oplus idG_2, id\}$, by $\{f \oplus e_i, id\}$and terminates the session. The new data stored is $\{f \oplus r'G_1 \oplus idG_2, id\}$.
- At session (j), R sends the query message with id_R to T.
- T searches $\{f \oplus r'G_1 \oplus idG_2, id\}$ corresponding to id_R. T generates a random error vector e_j and computes $y = r'G_1 \oplus idG_2 \oplus e_j$ and sends it to R.
- After decrypting y, the received *id"*,*r''* is different from id, r' (stored in the database). Thus, the tag's authentication has failed.

There is another scenario to realize the attack on desynchronization. When the intruder blocks the last message, the random value is updated in back-end and not modified in the tag. Consequentially, the tag and the reader are not correlated and this protocol does not achieve the desynchronization resilience property.

5.7.2 Performance analysis

Let u denote number of authorised readers with a tag. The space of stored memory of tag is depending on u, if u is important then we require $u\times(n+k_2)$ bits.

Other important factor is circulate matrix A_p, authors propose to calculate the circulate matrix A_p from vector p then compute eA_p , this requires a more complex computation and important space in the volatile memory ($n\times n$) bits.

7.8 Chien's Protocol (2013)

5.8.1 Review of Chien's Protocol

H-Y. Chien [Chi13] proposed RFID authentication protocol based on a combination between Rabin cryptosystem and error correction codes to achieve anonymity and untraceability proprieties. The author proposed two authentication protocols according to the security of communication between the reader and the server (secured/unsecured). In this paper, we are interest by the protocol in which the communication between reader and server is secured.

a) Initialization phase

Initially, R assigns $\{c, id, K, r\}$ to tag T, where c is one non-zero codeword, id is tag's identifier; K is shared key between S and T, and secret random value r. The server (reader) keeps $\{id, c, K, r_{old}$ and $r_{new}\}$ for each tag and public-key matrix G. r_{old} represents the r value used in the previous session, r_{new} represents the r value is used in the next session, and $r_{old} = r_{new} = r$ initially.

b) ***Authentication phase***

The authentication protocol is depicted as follows (Figure 5.9):

- R sends its query message with a random number N_R to T.
- T generates a random error vector e and computes $c_i=c+e$ and $V_T=g(e\oplus g(N_R\oplus K\oplus r))$. T calculates $M=(c_i\|V_T)^2 \ mod\ N$ and send it to T.
- R who knows two prime numbers first applies the Chinese reminder theory to derive four answers $\{c_i\|V_T\}$. For each answer, the reader decods c_i to get (c,e) to identifier of the corresponding tag.

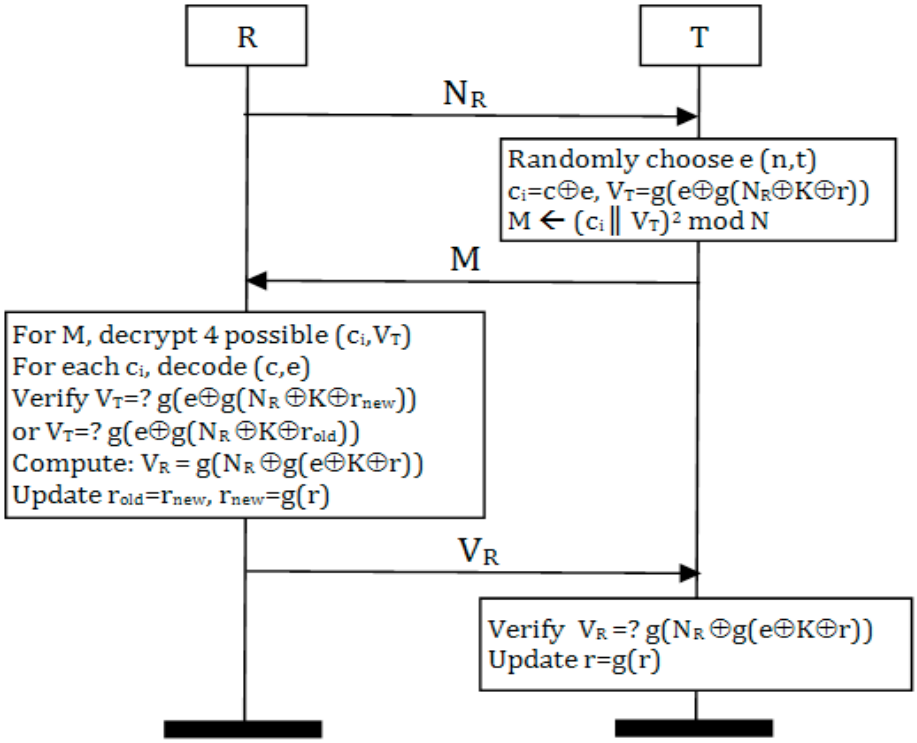

Figure 5.9: Chien's Protocol (2013) [Chi13]

- *R* computes and verifies whether V_T?=g(e⊕g(N_R⊕K ⊕r_{old})) or V_T?=g(e⊕g(N_R⊕K ⊕r_{new})). When this one is identified, the tag's authentication is successful.
- *R* computes V_R=$g(N_R$⊕$g(e$⊕K⊕$r)$).It updates r_{old}←r_{new} and r_{new}←$g(r_{new}$). The *R* sends V_R to *T*.
- *T* verifies the equality V_R?=g(N_R⊕ g(e⊕K⊕r)) .If successful, it accepts the reader's authentication, and updates r←$g(r)$.

5.8.1 Security analysis

This protocol is formally secured and it achieves the security and privacy proprieties, but this is not the only factor to evaluate a protocol. Author used Rabin cryptosystem to avoid message-resend attack and untraceability attack.

Using Rabin cryptosystem implicates adding a space memory and adding others computational operations (square modular and square root modular). We propose other solution; this principle is based on the dynamic of weight of error vector in each session wherever less then *t*.

Also, Author selects N=512 as size of public key of Rabin, but the size key 512-bit number is factored in 1999 by the Number Field Sieve factoring method (NFS). Actually, the size key recommended is 2048 bits. Among techniques used to resolve the problem of modular square root to determine the correct plaintext (4 plaintexts possible), we cite redundancy scheme, but the author of [Chi13] did not use this scheme, these implicates the decoding of codeword and compututaion four g four times.

The protocol used McEliece cryptosystem because it is very fast and resistant to quantum computer, but the Rabin (especially of RSA) is not fast relatively to McEliece and cannot resist quantum computer.

7.9 Li et al. Protocol

Li et al. proposed in [LYL14] a mutual RFID authentication based on the QC-MDPC McEliece cryptosystem. It was designed to secure mutual authentication and to resist replay attack.

5.9.1 Review of Li et al. Protocol

a) Initialization phase:

In the initialization phase, the trusted center (e.g. server) generates the initialization vector $h' \in F_2^n$, saves it in the tag T and the reader of R with identifier $id \in F_2^k$.

b) Initialization phase:

The scheme works as follows (see Figure 5.10):

- The reader R generates a random vector v and queries the tag T.
- After receiving the vector v, T randomly generates an error vector e, and then utilizes the vector h' to create public-key matrix G for encryption. Then, it computes $c'=idG \oplus e$ and $h_1=hash(p \| e)$, and sends c' and h_1 back to the reader.
- After receiving the authentication message from R and transmitting them to back-end database, R performs a decoding algorithm with private key matrices and identifies the error vector e as well as id. From id, the server retrieves the corresponding value of id.
- R computes $h(p \| e)$ and compares it with h_1. If they are equal, R computes $h_2=h(e)$ and sends it to T.

- *T* would compute $h(e)$, if $h(e)=h_2$, then the object of mutual authentication is achieved, authentication is successful, otherwise, the reader's authentication has failed.

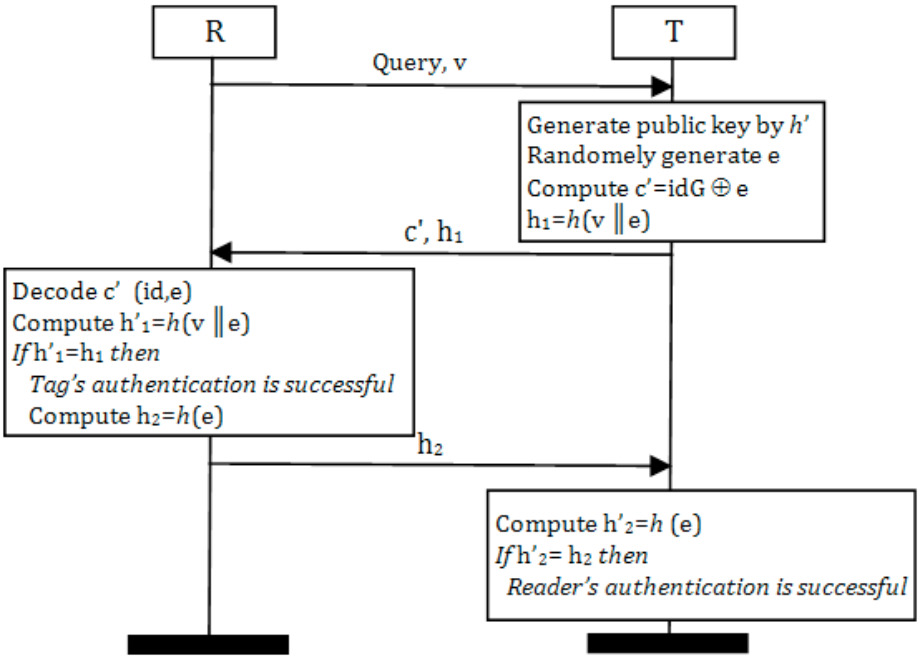

Figure 5.10: Li et al.'s Protocol [LYL14].

5.9.2 Traceability attack

In the McEliece cryptosystem, the parameters (n,k,t) are public. With these information, and particularly, the minimum distance d and the Hamming weight t; the adversary can attempt to trace the tag with the following scenario:

- At session (i), the adversary intercepts $(c'_i=idG\oplus e_i)$ and saves it.
- At session (j), it intercepts $(c'_j=idG\oplus e_j)$.
- The intruder computes: $c'_i\oplus c'_j=idG\oplus e_i\oplus idG\oplus e_j$

We have $e_i\neq e_j$ and the identifier of the tag *id* is static in all sessions, this implicates: $c'_i\oplus c'_j=e_i\oplus e_j$. The Hamming weight of $(c'_i\oplus c'_j)$ is less than $2t+1$, and the codeword idG is fixed for all sessions which leads to message-resend attack, and implicates, that this protocol does not provide untraceability.

5.9.3 Violation of forward secrecy

If an intruder compromises a tag, then it might be able to derive previous secret data to track old transactions involving that tag, thus violate forward secrecy. In Li et al.'s protocol, the data stored in the tag's memory are {*id*, *h'*}, which remain constant in all the runs of protocol. An intruder breaking into the memory of tag gets the current *id*. The problem posed is that the value of the identifier is static and not dynamic. Therefore, this protocol does not achieve forward secrecy.

5.10 Conclusion

In this chapter, we have analysed different RFID authentication protocols. Theses studied protocols require error-correcting codes for assuring security requirements (tag's authentication, reader's authentication, untraceability, etc.).

In next chapter, we will propose improved version of two recent protocols, Malek-Miri protocol [MM12] and Li et al. protocol [LYL14].

Chapter 6

Improved Code-based RFID Authentication Protocols

6.1 Introduction

In this chapter, we propose two improved RFID mutual protocols using two code-based schemes, the first one is based on the randomized McEliece cryptosystem [CCCB15b] and the second one is based on Quasi Cyclic-Moderate Density Parity Check (QC-MDPC) McEliece cryptosystem [CCCB15c].

We provide security properties using AVISPA (Automated Validation of Internet Security Protocols and Applications) tools [ABBC+05]. We use the privacy model of Ouafi and Phan [OP08] to verify the untraceability property. Our work also includes a comparison between our improved protocols and different existing code-based RFID authentication protocols in terms of security and performance.

6.2 RFID authentication protocol based on randomized McEliece cryptosystem (R2McE)

We propose in this section an improved RFID mutual authentication protocol using code-based scheme. Our protocol is based on randomized McEliece cryptosystem (*R2McE*), and uses an efficient decoding/encoding algorithm to generate an error vector of fixed weight. The only datum stored in tag is a dynamic identifier, and it is updated before the end of the session and without the need to do exhaustive search to obtain the identifier from database. This protocol is published in [CCCB15b].

6.2.1 System Model

The RFID system consists of three entities: tag *T*, reader *R* and server *S*.

- The tag *T* is low-cost and passive. It stores the dynamic identity (*DID*) which is strictly confidential. *T* implements an application ϕ (this application is discribed in section 2.5.2 of chpater 2) and pseudo-random numbers generator (PRNG) to generate *x* and compute *g*(.). It also supports bitwise operations (xor, and,...). A tag has a rewritable memory that may not be tamper-resistant.
- The reader *R* can generate pseudo-random numbers.
- The server *S* has a sufficient storage space and computational resources. We implement algorithms of ϕ and PRNG. Server *S* can decode the message received from *T*, then, we implement encryption/decryption of randomized McEliece cryptosystem with public-key matrix *G*, private-key matrices and a polynomial-time decoding algorithm $\psi(.)$. The server contains the database which includes $\phi_{n,t'}$.

In our work, we propose to use $\phi_{n,t'}$ as follows (Algorithm 6.1):

Algorithm 6.1 Generation an error vector
Randomly choose $x \in \left[0, \binom{n}{t}\right]$ **repeat** determinate the largest *t'* such that $x \in \left[0, \binom{n}{t'}\right]$ **until** t'<t **compute** $\phi_{n,t'} = e$ where wt(e) =t'<t

We will choose *t'* such that the syndrome decoding problem (most efficient algorithm) remains hard.

6.2.2 Description of R2McE protocol

Our proposed Protocol R2McE is divided into two phases: the registration phase and the mutual authentication phase.

a) ***Registration phase***

The server generates a random binary Goppa code $C[n,k,d]$ as specified by the generator matrix G', where $G=S'G'P$ and G is public-key. The server S generates random values using PRNG, id the unique identifier of tag and the random number r. It computes $c_r = rG_1$, $c_{id} = idG_2$, and $DID = c_r \oplus c_{id}$. , and initializes $c_{r_{old}}$ and $c_{r_{new}}$ by c_r. Then, the server (registration center) sends *DID* to the tag through a secure channel, where *DID* is strictly confidential. S stored in the database $\{id, c_{id}, c_{r_{old}}, c_{r_{new}}\}$ for each tag.

b) ***Mutual authentication phase***

The mutual authentication phase is described as follows (and in Figure 6.1):

Step 1. ***Tag's Authentication***

Step 1.1. R generates a nonce and sends it as a request to the tag T.

Step 1.2. T generates a random number $x \in [0, log_2\binom{n}{t}[$ and $t' \in [1, t[$, and calculates error vector e with $wt(e)=t'$ from $\phi_{n,t'}$, $c'=DID \oplus e$ and $P=g(N_R \| x \| DID)$.

Step 1.3. T sends c' with P to the reader, and resends the received c', message P and nonce N_R to the server S.

Step 1.4. S performs a decoding algorithm $\psi(.)$ with private key matrices and identifies the error vector e as well as id and r. From id, in database, the server retrieves the values of $c_{id}, c_{r_{old}}, c_{r_{new}}$ and calculates $\phi_{n,t'}^{-1}(e)$ and $P_1=g(N_R \| x \| (c_r \oplus c_{id}))$ (either $c_{r_{old}}$ or $c_{r_{new}}$). S verifies if $P_1 \stackrel{?}{=} P$, if they are equal, the tag's authentication is successful; otherwise the tag's authentication has failed.

Step 2. ***Reader's Authentication***

Step 2.1. In the case of the tag's authentication is successful, the server generates a nonce r' and computes $c_{r'}=r'G_1$ and $DID_{New} = c_{r'} \oplus c_{id}$. It computes

$Y=DID_{New}\oplus e$ and $Q=g(N_R \| DID_{New} \| x)$. It updates $c_{r_{old}} \leftarrow c_{r_{new}}$ and $c_{r_{new}} \leftarrow c_{r'}$, only in case the matched c_r is $c_{r_{new}}$.

Step 2.2. *S* sends *Y* and *Q* to the reader and resends the received message to *T*.

Step 2.3. *T* obtains DID_{New} by calculating $Y\oplus e$ and calculates $Q_1=g(N_R \| DID_{New} \| x)$. *T* verifies if $Q_1 \stackrel{?}{=} Q$, if they are equal, the reader's authentication is successful; otherwise the authentication of the reader has failed.

Step 2.4. *T* updates the dynamic identifier by the value of DID_{New}, if reader's authentication is successful.

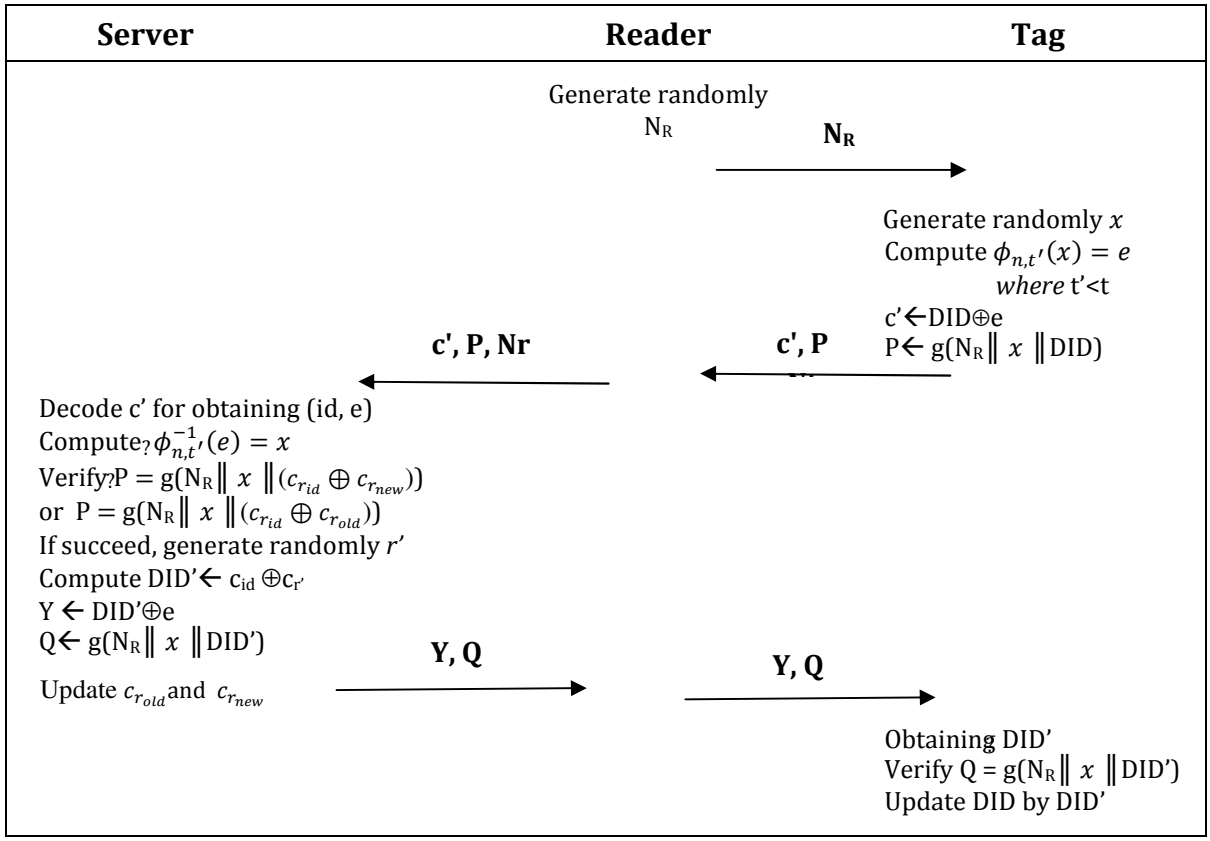

Figure 6.1: Our improved protocol - R2McE

6.2.3 Automated verification

Using AVISPA tools [BBBC+05], we verify the secrecy and mutual authentication security properties of R2McE.

Our protocol R2McE requires the primitives: PRNG, nonce xor-operator and McEliece cryptosystem. The randomized McEliece cryptosystem requires the primitives:

public key, private key, application ϕ and the decoding algorithm $\psi(.)$ which is used with a private key to obtain *id* and *e*. The application ϕ is bijective, but the intruder cannot find *x* without knowing the value of *t*', and the result of this application *e* does not circulate clearly in the channel, then we can model it by a hash function *Phi*(*x*). The intruder will know this function, therefore he will be able to compute the error vector but not invert values of $Phi^{-1}(x)$ (unless he already knows *x*).

Concerning the message $DID \oplus e$, we cannot specify it in HLPSL by *xor*(*DID*,*E*) because the reader does not use the algebraic properties of or-exclusive operator (e.g. neutral element) to obtain *id* and *e*. To retrieve these values, we apply the private decoding algorithm $\psi(.)$ and the private key of McEliece. $DID \oplus e$ means the encoding *DID* by *e*, where *DID* is encryption of $[r \| id]$ by public key *G*. The reader (server) obtain the value *DID* and *e* uses the private decoding algorithm $\psi(.)$. Therefore, we propose to specify this message in HLPSL by {*DID*}_*E*. The other side, we can specify the message $DID_{New} \oplus e$ by *xor*(*DNew*,*E*) (last message from reader to tag) because the objective of the tag is to retrieve the value of DID_{new} using the algebraic properties of xor operator.

The Appendix *D* shows the specification of R2McE protocol by HLPSL. In our protocol, the honest participants are the reader *R* and the tag *T*. Then, we have two basic roles, the *tag* and the *reader*. We can define a session role where all the basic roles are instanced with concrete arguments. In the *tag*, we initialise the argument *DID* by {ID.Rinit}_kG. In the *reader*, we initialize the values *Rold* and *Rnew* by *Rinit*. We provide a validation of properties: the tag's authentication (*aut_tag*), the reader's authentication (*aut_reader*), the secrecy of current *DID* (*sec_did*1), and the secrecy of the new *DID* (*sec_did*2).

The Figure 5.2 shows the result of verification of our protocol by AVISPA tools. This result clearly means that there is no attack detected (replay or man-in-the-middle attacks). We can thus deduct that the diagnostic of AVISPA tools for our protocol is secure.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  UNTYPED_MODEL
PROTOCOL

C:\progra~1\SPAN\testsuite\results\R2McE.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 2543 states
Reachable  : 325 states
  Translation: 0.04 seconds
  Computation: 0.18 seconds
```

Figure 6.2 : Verification result using CL-AtSe tool of R2McE protocol

6.2.4 Privacy verification

We use the Ouafi-Phan model to verify the achievement of untraceability property in our R2McE improved protocol. At session (*i*), by the *Execute* query, the adversary $\mathcal{A}$ eavesdrops a perfect session between T_0 and a legitimate reader. It obtains the values $DID_i \oplus e_i$ and $g(N_{Ri} \| x_i \| DID_i)$. At next the session, an intruder cannot replay a previously used $g(N_R \| x \| DID)$ and $DID \oplus e$ to a reader, even with high probability, it will not match the N_R value generated by the reader for that session. There are two mechanisms to against the replay. Firstly, generating an error vector with dynamic length $t' \leq t$ where t' is confidential. Secondly, accepting the principle of dynamic codeword, which is stored in tag in the form of *DID*. In each session, the transmitted encoding codeword is different from the codeword of the last session because the value of the codeword is updated in the server and in the tag before the end of the session.

In addition, the security of our protocol is based on security of randomized McEliece. Nojima et al. [NIKM08] prove that padding the plaintext (in our protocol, identifier of tag *id*) with a random bit-string (random number *r*) provides the semantic security against chosen plaintext attack (*IND-CPA*) for the McEliece cryptosystem with the standard assumptions. So, the randomized McEliece cryptosystem is *IND-CPA* secure, which means

that no probabilistic polynomial-time adversary wins the *IND-CPA* experiment with an advantage greater than a negligible function of the security parameter.

6.2.5 Performance evaluation

The performance of authentication protocols is mainly measured by storage space on tag and computation cost in tag and server, and communications cost between the tag and the reader.

The storage space Concerning the space required in tag's memory, our R2McE protocol requires to store only datum that is dynamic identifier *DID*, whose length is *n* bits, where *n* is length of codeword.

The computation cost, the tag requires simple operations: pseudo-random number generator (PRNG), or-exclusive operation, and application$\phi_{n,t\prime}$. We used the PRNG to generate *x* and to compute *g*(.), which proved to be very fast. The cost of application ϕ is $O(t\ell^2)$ binary operations.

If we select a binary Goppa code *C*[*n*=2048,*k*=1751,*d*=56], these parameters is suitable with the parameters of a secure McEliece cryptosystem for 2^{80} security [BLP08]. We choose the values of k_1= 890 and k_2= 875 which are suitable with condition $k_2<k_1$. So, the number of tags supported is 2^{875} tags and the space memory required in the tag is 2048 bits for codeword *DID* and the maximal weight of the error vector is 27 bits. With these parameters, we can implement R2McE protocol in low-cost tags, such as Mifare Classic 1K and Mifare Plus support space memory 1KB to 4 KB [Mif].

The communication cost between a tag and a reader consists of: the number of message exchanges, and the total bit size of the transmitted messages, and this per each communication. As for R2McE protocol, the total of the bits of the messages of communication is $2n + 3l_p$, where l_p is length of random number generator.

6.3 Our RFID authentication protocol based on QC-MDPC McEliece cryptosystem (RQMcE)

6.3.1 System model

- The tag *T*: In our context, it is passive and it stores {*id*, *rand*, *h'*} which are strictly confidential. *T* implements key generation algorithm and encryption algorithm of

QC-MDPC cryptosystem. It also implements pseudo-random number generator and supports bitwise operations (xor, and,etc.).

- The reader *R*: It can generate the pseudo-random numbers with a PRNG.
- The server *S*: We implement decryption algorithm of QC-MDPC cryptosystem and PRNG. It contains the private-key and the database which includes {*id,* $rand_{old}$, $rand_{new}$}.

6.3.2 Description of RQMcE protocol

The proposed protocol is divided into two phases: the initialization phase and the authentication phase.

a) ***Initialization phase***

In this phase, the tags and the database server are initialized for authentication process to be performed in the future. The server generates a random binary QC-MDPC code $\mathcal{C}(n,r,w)$. The server (trusted center) generates the initialization vector $h' \in F_2^n$, the unique identifier of tag $id \in F_2^{k_2}$ and shared secret $rand \in F_2^{k_1}$. Then, the server sends {*id*, *rand*, *h'*} to the tag through a secure channel. It stores in the database {*id*, *rand*} for each tag and *h'*, where $rand = rand_{old} = rand_{new}$.

b) ***Mutual authentication phase***

The mutual authentication phase takes place as follows (to see Figure 6.3):

Step 1. ***Tag's Authentication***

Step 1.1. *R* generates a nonce N_R and sends it then as a request to the tag *T*.

Step 1.2. *T* generates an error vector *e* with $wt(e) \leq t$, and computes $c' = [rand \| id]G \oplus e$. It also computes $U = g(id \| N_R \| e)$.

Step 1.3. *T* sends *c'* with *U* to the reader, it resends the received *c'* and message *U* and nonce N_R to the server.

Step 1.4. The server runs decryption algorithm to find *id*, *rand* and *e*. From *id*, in database, the server obtains the values of $\{rand_{old}, rand_{new}\}$. if $rand=rand_{old}$ or $rand=rand_{new}$ then the tag computes $U_1=g(id \| N_R \| e)$ (either $rand_{old}$ or

$rand_{new}$) and verifies if $U_1 \stackrel{?}{=} U$. If they are equal, authentication of tag is successful; otherwise the authentication of tag has failed.

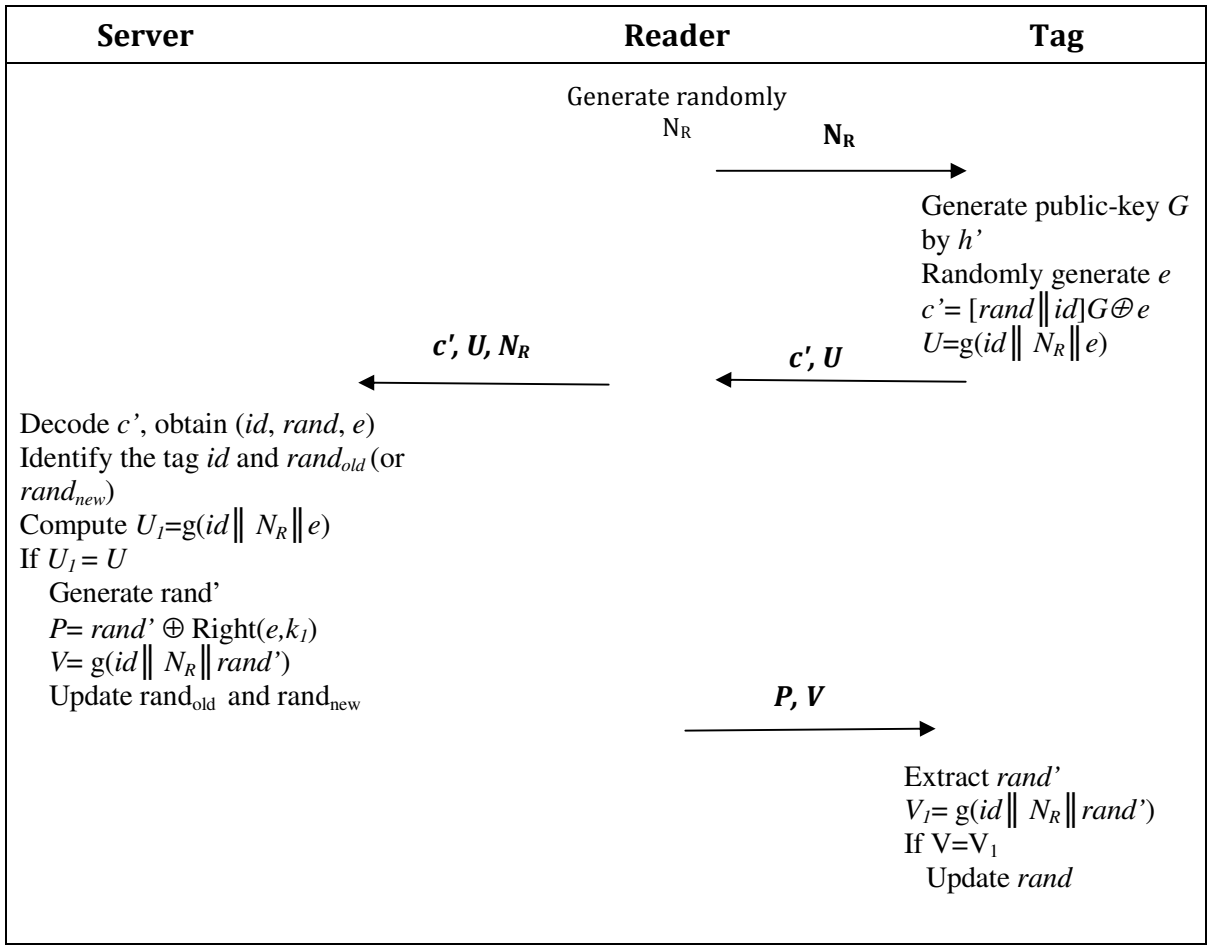

Figure 6.3: Our improved protocol - RQMcE

Step 2. ***Reader's Authentication***

Step 2.1. In this case the authentication of tag is successful. The server generates a random number $rand \in F_2^{k_1}$ and computes $V=g(id\|N_R\|rand')$ and $P = rand' \oplus Right(e, k_1)$. It updates $rand_{old} \leftarrow rand_{new}$ and , only in case the matched *rand* is $rand_{new}$.

Step 2.2. *S* sends *P* and *V* to the tag.

Step 2.3. *T* obtains *rand*' by computing $P \oplus Right(e, k_1)$. It computes $V_1=g(id\|N_R\|rand')$ and checks if $V_1 \stackrel{?}{=} V$. If they are equal, the authentication of reader is successful; otherwise the authentication of the reader has failed.

Step 2.3. *T* updates the secret *rand* by the value of *rand*', in case of the reader's authentication is successful.

6.3.3 Automated verification

The RQMcE protocol requires the primitives: PRNG, nonce, xor-operator, public-key, private-key and encryption/ decryption of Randomized McEliece cryptosystem based on QC-MDPC codes. We have two honest agents tag and reader. We can present the ciphertext $c'=[id\|rand]G\oplus e$ as $F_{Encry}([id, rand], PKG, E)$ that means encryption $[id\|rand]$ by public-key *PKG* (is matrix *G*), then encoding the result by the private error vector *E* (is *e*). To obtain the value of *E*, one uses the decoding algorithm ψ_H. So, The specification of this ciphertext by HLPSL is *{{Rand.ID}_PKG}_E*. We specify the functions *g*(.) and *Right*(.) by hash function. Other primitives are defined in HLPSL.

We define a *session* role where all the basic roles are instanced with concrete arguments. In the *reader*, we initialize the values $Rand_{old}$ and $Rand_{new}$ by *rand*.

We provide a validation of properties: authentication of tag (*auth_tag*), authentication of reader (*auth_reader*), the secrecy of identifier of tag *id* (*sec_id*), and the secrecy of secret random number *rand* and the new random number *rand*' (*sec_rand* and *sec_randp*). These properties are specified in ***goal***.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  UNTYPED_MODEL
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\RQMcE.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 2750 states
  Reachable  : 1696 states
  Translation: 0.01 seconds
  Computation: 0.39 seconds
```

Figure 6.4: Verification result of RQMcE protocol

We also provide that our scheme resists to replay attack and man-in-the-middle attack. HLPSL specification of our improved scheme is shown in Appendix *E*.

The result of verification of our protocol by AVISPA tools is presented in Figure 6.4. This result clearly means that there is no attack detected. We can thus deduct that the diagnostic of AVISPA tools for our protocol is secure.

6.3.4 Privacy verification

Using Ouafi-Phan model, we validate the untraceability property in our RQMcE protocol. At session (*i*), by the *Execute* query, the adversary $\mathcal{A}$ eavesdrops a perfect session between T_0 and a legitimate reader. He obtains the values $[randi \| idi]G \oplus ei$ and $g(idi \| N_{Ri} \| ei)$. At next session, the intruder cannot replay a previously used $[rand \| id]G \oplus e$ and $g(id \| N_R \| e)$ to a reader, since with high probability, it will not match the N_R value generated by the reader for that session.

On the other side, we apply QC-MDPC McEliece cryptosystem with padding the plaintext by a random bit-string where the exchanged encoding codeword is different in each session. In RQMcE protocol, we have two messages in two different sessions:

$$, \text{where } c_i = [\text{rand}_i \| id]G$$

and

$$c'_j = c_j \oplus e_j, \text{ where } c_j = [\text{rand}_j \| \text{id}]G$$

where $c_i \neq c_j$ and $e_i \neq e_j$. The intruder intercepts c'_i and c'_j as follows:

$$c'_i \oplus c'_j = c_i \oplus c_j \oplus e_i \oplus e_j,$$

In case $wt(e_i)=wt(e_j)=t$ and $c_1=c_2$ or the adversary knows the linear relation between the messages m_i of c_1 and c_2 then this protocol does not resist traceability attack.

In our protocol, the vector $rand_i$, which is used in session *i* is different from $rand_j$ which is used in session *j*, and there is no linear relation between them, $rand_i$ and $rand_j$ are randomly generated. We note that $wt(e_i)$ and $wt(e_j)$ are secret and different. Then, our scheme resists traceability attack.

6.3.5 Performance evaluation

Storage cost The improved protocol requires {*id*, *rand*, *h*} with size $k+n$. The QC-MDPC code $C[n=9602,r=4801,w=90]$, $n_0=2$ and $t=84$ are parameters proposed by Misoczki et al. [MTSB13] for a 2^{80} security. Using these parameters, the memory space requires in the tag are 14403 bits ($n+r$). If we choose $k_1=4300$ and $k_2=501$ which is suitable with condition $k_1<bk$ and $b=9/10$ then, we can implement our scheme in low-cost tags, such as Mifare Classic 1K and Mifare. The number of tags which can be use in our protocol is 2^{501} tags.

Calculation cost RQMcE protocol requires QC-MDPC McEliece cryptosystem with padding of the plaintext by a random bit-string, PRNG and xor operation. The QC-MDPC McEliece cryptosystem is designed to reduce the key sizes [MTSB13]. The works of [HMG13, MG14] present a very lightweight implementation of the QC-MDPC McEliece cryptosystem for embedded devices. We used the PRNG to generate $\{N_R, e\}$ and compute $g(.)$, which is very fast. We also cite that, the server does not need an exhaustive search to obtain the value of *id*. When the server decrypts the encoded codeword, it can obtain the value of tag's identifier.

Communication cost The total of the bits of the messages of communication in authentication process is $3l_p+n+k_1$, where k_1 is the length of random number *rand*.

6.4 Security Comparison

A secure RFID authentication protocol should provide mutual authentication, secrecy, untraceability, desynchronization resilience, forward secrecy and replay attack resisting. In this section, we discuss the security and privacy requirements of our proposed protocols and others protocols. Table 6.1 presents the security comparison between the existing protocols and our proposed protocols.

6.4.1 Mutual authentication

If the RFID protocol is successfully achieved, tag authentication and reader authentication is successful too, then one can say that this protocol is providing mutual authentication. The protocols proposed in [Par04, Chi06, CKMI07, SKI10] are one-way authentication protocols, thus they don't achieve the reader (or server) authentication. We have verified the achievement of mutual authentication in our proposed protocols by AVISPA tools.

6.4.2 Secrecy

In all studied protocols and also in our proposed protocols R2McE and RQMcE, the tag's identifier and secret information are secured. These data are protected by a code-based encryption scheme: McEliece and its variants and Niederreiter and its variants. In our proposed protocols, this property is verified by AVISPA tools.

6.4.3 Untraceability

The weight of error vector in protocols [Par06, CL09] is fixed, when the intruder knows d or t then it can follow the trace of the tag. To achieve the property of untraceability, we have proposed two mechanisms: dynamic weight and dynamic codeword. The first one is by generating an error vector with dynamic weight $t' \leq t$ where t' is confidential. The last one is by agreeing on the the principle of dynamic codeword, which is stored in tag in form dynamic identifier *DID* in case of [CCCB15a] and add a random padding number in each new session in our protocol [CCCB15c]. In each session, the transmitted encoding codeword is different from the codeword of the last session because the value of, the codeword is updated in the server and in the tag before the end of the session. We prove that our proposed protocols achieve untraceability property by Ouafi-Phan model.

6.4.4 Desynchronization resilience

The secret information shared between tag and reader (or server) in protocols [Par04, Chi06, MM12] are dynamic and are not protected by the technique of secret desynchronization, thus these protocols do not resist desynchronization attacks. However, the secret information in protocols [CKMI07, CL09, SKI10, LYL14] which are stored in tag's memory are static in all sessions, then the problem of desynchronization attack is not posed for these protocols . In R2McE and RQMcE protocols, the random value in codeword is updated in each session. Therefore, to achieve this property, we stored two secret synchronisation information in the server, $(c_{r_{old}}, c_{r_{new}})$ for R2McE protocol, and (r_{old}, r_{new}) for RQMcE protocol. Then, our two proposed protocols resist desynchronization attack.

6.4.5 Forward secrecy

In protocols [CKMI07, CL09, SKI10, LYL14], the information stored in the tag's memory remain static in all the runs of scheme. An intruder breaking into the memory of the tag gets the current *id*. The problem posed is the value of identifier when static and not

dynamic. Concerning our proposed protocols, before termination of the session, the tag updates the value of the secret information, *DID* in R2McE protocol and *rand* in RQMcE protocol. The adversary could not acquire the previous random vector *rand* used in the prior sessions. So, our proposed RFID authentication protocols could provide forward secrecy.

	M.A	D.C	Unt	D.R	F.S	R.R
Park [Par04]	N	Y	N	N	Y	Y
Chien, 06 [Chi06]	N	Y	Y	N	Y	Y
Cui et al. [CKMI07]	N	Y	Y	Y	N	Y
Chien-Laih[CL09]	Y	Y	N	Y	N	Y
Sekino-al [SKI10]	N	Y	Y	Y	N	Y
Malek-Miri [MM12]	Y	Y	Y	N	Y	Y
Chien, 13 [Chi13]	Y	Y	Y	Y	Y	Y
Li et al. [LYL14]	Y	Y	N	Y	N	Y
RQMcE	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
R2McE	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**

M.A: Mutual Authentication, *D.C*: Data Confidentiality

Unt: Untraceability, *D.R*: Desynchronization resilience

F.S: Forward secrecy, *R.R*: Resist replay attacks

Table 6.1: Comparison of security and privacy properties

Remark We note that our proposed protocols as well as Chien's protocol [Chi13] have proved security and privacy properties, though our protocols are based only on error-correcting codes, it is better in performance analysis (storage space and computation cost), view details in Table 6.2.

6.5 Performance Comparison

The performance of authentication protocols is mainly measured by storage space on the tag, computation cost in tag and server and communications cost between the tag and the reader. Our comparison is articulated on authentication phase for each protocol.

The performance comparison between our proposed protocols and the existing code-based RFID protocols in terms of storage cost and computation cost is summarized in Table 6.2.

6.5.1 Storage cost

Concerning the storage cost, the tags in protocols [CKMI07, SKI10] require public-key matrix which is of important size compared to resources of low-cost tags. The data stored on tags of protocol [Par04, Chi06] are multiple in an agreed number of sessions and in [MM12] multiple in number of authorized readers. The protocol of [Chi13] requires important space for the *id*, symmetric-key, public-key of Rabin cryptosystem and unique codeword. R2McE protocol requires n bits for dynamic identifier *DID*. The RQMcE protocol requires k bits for vector h and n bits for {*id*, *rand*}. Then, the sum is $k+n$. Thus the space requiring in our proposed protocols R2McE and RQMcE are compatible whith resources of low-cost tags.

	Key space	*Computation*	
		Tag	*Server*
Park [Par04]	$l_p+n+2\|key\|$	*1P*	*iP+1D+1ED*
Chien, 06 [Chi06]	$n+ l_p+ \|key\|$	*(n'-i+1)P*	*(n'-i+1)P+ 1ED*
Cui et *al.* [CKMI07]	$(n-k)\times(n_2+1)$	*2P + 1EC*	*4P + 2ED*
Chien-Laih [CL09]	$n+2\|key\|$	*8P*	*2P + 1ED*
Sekino et *al.* [SKI10]	$(n-k)+(n-k)\times(n_1-(n-k)/t$	*1EC + 2P*	*2P + 1ED*
Malek-Miri [MM12]	$(n+k_2+\|key\|)$	*2P + CM*	*2P + 1ED*
Chien, 13 [Chi13]	$n+ \|N\|+3\|key\|$	*1 SQ + 6P*	*10P+1SR+4ED*
Li et al. [LYL14]	$(n+k_2+\|key\|)$	*3P + GG*	*2P + 1ED'*
RQMcE	$n+k$	*3P+ GG*	*2P + 1ED'*
R2McE	n	*3P*	*2P + 1ED*

$|key|$: length of *key* or *id* l_p: length of generating random number or hash.
i: number of authorised sessions *P*, *D* and *CM*: cost of RNG or hash function, decryption operation
GG: cost of generation of matrix *G* and generation of circular matrix, respectively.
EC, *ED, ED'*: encoding operation, decoding operation of McEliece, decoding operation of QC-MDLP with McEliece, respectively.
SQ and *SR*: cost of squaring and square root solving, respectively
$|N|$: public-key of Rabin cryptosystem

Table 6.2: Comparison of space and computation costs

6.5.2 Computation cost

As for the computation cost, the main advantage in all code-based RFID authentication protocols in relation to hash-based RFID authentication protocols is that

there is not need of exhaustive search to obtain the value of tag's identifier. In addition, The McEliece cryptosystem (also other its variants) is of high-speed encryption and decryption compared to asymmetric cryptosystems based on number theory, such as Elliptic Cube Cryptosystem (ECC) and ELGamal cryptosystem. The low-cost tags require simple operations: pseudo-random number generator and xor operations.

With regard to the other protocols and consideration of mutual authentication, the performance of our proposed protocols is effective. We mention here an important remark, in the MQMcE protocol, in each session the tag generates a public-key from the stored vector h and applies encryption algorithm to encryption $[id \| rand]$. This protocol is based on QC-MDLP cryptosystem which can implement it in embedded devices, like in [HMG13, MG14].

6.5.3 Communication cost

We evaluate the communication cost by the amount of bits of transmitted messages in the RFID protocol from tag to reader and in vice versa. All nonces are generated by PRNG with length l_p. The length of ciphertext of McEliece cryptosystem and its variants is n and length of ciphertext of Niederreiter cryptosystem and its variants is $(n\text{-}k)$. Table 6.3 shows the comparison between our proposed protocols and the existing RFID protocols based on error-correcting codes in term of communication cost.

	$T \rightarrow R$	$R \rightarrow T$	**Sum**
Park [Par04]	n	-	n
Chien, 06 [Chi06]	n	l_p	$n+l_p$
Cui et *al.* [CKMI07]	$(n\text{-}k)+l_p$	l_p	$(n\text{-}k)+2l_p$
Chien-Laih[CL09]	$2\ l_p+2n$	$2\ l_p$	$4\ l_p+2n$
Sekino et *al.* [SKI10]	$(n\text{-}k)+l_p$	l_p	$(n\text{-}k)+2\ l_p$
Malek-Miri [MM12]	n	$2n+\|key\|+l_p$	$3n+\|key\|+l_p$
Chien, 13 [Chi13]	$\lceil N \rceil$	$2\ l_p$	$2\ l_p+\lceil N \rceil$
Li et al. [LYL14]	$n+l_p$	$2\ l_p$	$n+3l_p$
RQMcE	$n+l_p$	$k_1+2\ l_p$	$n+3l_p+k_1$
R2McE	$n+l_p$	$n+l_p$	$2n+l_p$
$\|key\|$: length of *key* or *id*		l_p: length of generating random number or hash	

Table 6.3: Comparaison of communication cost

In our proposed protocols is less than the number of bits in protocols of [CL09, MM12]. On other side, it is greater than the number of bits in protocols of [Par04, CKMI07, SKI10, LYL14]. If we consider the importance of the factor of security depending on communication cost, we can conclude that R2McE and RQMcE are effective.

6.6 Conclusion

In this chapter, we have proposed two improved RFID authentication protocols based on two variants of McEliece cryptosystem with mutual authentication, untraceability, desychronisation relisience and forward secrecy. Using privacy model of Ouafi-Phan and AVISPA tools, we have proved the security and privacy properties.

With regard to the different existing protocols based on error-correcting codes, the performance of our proposed protocols are effective, the space memory required is compatible with available space on the low cost tag, they do not need to do exhaustive search, and the tag can perform lightweight cryptographic operations.

Conclusion and perspectives

The subject of this thesis is the study of the security problems in embedded systems. This research domain is very vast; therefore we articulated our study on design and verification of authentication protocols as the security problem and the RFID system as an embedded system.

In topic of RFID security, we found many proposed protocols and each protocol has advantages and disadvantages in terms of security and performance. The main design objectives of a new authentication protocol in RFID systems are minimizing cost, development of performance, and validation of security and privacy properties. This equation is not validate in all proposed protocols.

Along our work, we concentrated our study on the security analysis and the performance analysis of recently proposed RFID authentication protocols. We can discover weaknesses in several protocols. These protocols are divided into two families, hash-based protocols and code-based protocols. In the first category, we verified two recent protocols [WHC11, JDTL12] by AVISPA tools. We showed that the two verified protocols cannot resist algebraic replay attack (ARA) on authentication, and therefore an intruder can impersonate the tag. The principal cause of the described attacks in our work is the misuse of the xor operator in the transmitted messages. The principal cause of the described attacks is the abuse of the proprieties of or-exclusive (xor) operator in the transmitted messages. We generalized these results to detect this type of attack in other protocols. Therefore, we have proposed a solution for this attack which is correcting the use of xor-operator and replacing it by the concatenation operator.

Using these results, we proposed a new authentication protocol (RBioA protocol) which is based on the combination between two systems, RFID system and biometric system, to apply it in access control applications, we used the principal of hash-based scheme to realize the security of protocol; used hash functions are cryptographic and biometric hash functions. The advantage of RBioA protocol is that it can be used in decentralized applications since we have no need of biometric database of the users in the system. Still, there is the problem of exhaustive research of tag's identifier in the server.

Other studied category of RFID protocols is code-based RFID authentication protocols (presented in chapter 5). Among these protocols, we provide enough evidence to

prove that two recent RFID authentication protocols [MM12, LYL14] are not secure. The results of security analysis showed that Malek-Miri authentication protocol [MM12] is vulnerable to desynchronization attack and Li et al.'s protocol [LYL14] does not provide untraceability and forward secrecy.

In chapter 6, we proposed the improved version protocols to prevent the described attacks. These protocols (R2McE and RQMcE) are based on two variants of McEliece cryptosystem. Using privacy model of Ouafi-Phan, we have proved the untraceability property. We verified the security properties by AVISPA tools. With regard to the different existing protocols based on error-correcting codes, the performance of our R2McE and RQMcE protocols are effective, does not need to do exhaustive search, and the tag can perform lightweight cryptographic operations.

Our perspectives of research include:

- Future research includes additional work in regards to the biometric hash function. There are many researches on the implementation of the robust hash function in RFID tags; but those on the implementation of biometric hash function are limited.
- We studied the RFID systems as independent systems. In a new technology, the components of RFID systems are communicated with other objects via different types of connection. This technology is called Internet of Things (IoT). Therefore, one need to propose a new approach to secure devices and systems of IoT and that takes in consideration their features.
- The better variant of McEliece cryptosystem used in our protocols and in existing protocols is of security IND-CPA, randomized McEliece cryptosystem. There is a problem if one wants to use the variant IND-CCA2 because it requires important resources, memory and computation.

Bibliography

[ABB+05] A. Armando, D. Basin, Y. Boichut, Y.Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.-C. H_eam, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santos Santiago, M. Turuani, L. Viganò, and L. Vigneron, "The AVISPA tool for the automated validation of internet security protocols and applications," *in Proceedings CAV 2005* (K. Etessami and S.K. Rajamani, eds.), vol. LNCS 3576, pp. 281–285, 2005.

[Abo10] H.A. Aboalsamh. "A Potable Biometric Access device using Dedicated Fingerprint Processor", *WSEAS Transaction on Computers*, Issue 8, Vol. 9, pp. 878-887, 2010.

[ADO06] G. Avoine, E. Dysli, and P. Oechslin, "Reducing time complexity in RFID systems," *in Proceedings of TCC 2009* (B. Preneel and S.E. Tavares, eds.) vol. LNCS 5444, pp. 474-495, Springer, 2009.

[AFS05] D. Augot, M. Finiasz, and N. Sendrier, "A family of fast syndrome based cryptographic hash functions," *in Proceedings of Mycrypt 2005* (E. Dawson and S. Vaudenay, eds.), vol. LNCS 3715, pp. 64–83, Springer, 2005.

[AlJ97] A. K. Al Jabri, "A Symmetric Version of the McEliece Public-Key Cryptosystem," *Int. J. Network Mgmt.,* vol. 7, pp. 316–323, 1997.

[BAN89] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," *Proc. Cambridge Phil. Soc.*, 1989.

[BCGO09] T. P. Berger, P-L. Cayrel, P. Gaborit, and A. Otmani, "Reducing key lengths with QC alternant codes," *in Proceedings of Africacrypt 2009*, vol. 5580, pp. 77–97, Springer, 2009.

[BDPR98] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," *in Proceedings of in Proceedings of Crypto 98*, pp. 26–45, 1998.

[Ber97] T. Berson, "Failure of the McEliece public-key cryptosystem under message-resend and related-message attack," *in Proceedings of Crypto 97*, 1997.

[BLP08] D.J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Proceedings of PQCRYPTO'08*, vol. LNCS 5299, pp. 31–46, Springer, 2008.

[BLPS11] D. J. Bernstein, T. Lange, C. Peters, and P. Schwabe, "Really Fast Syndrome-Based Hashing," *Cryptology ePrint Archive: Report 2011/074*, http://eprint.iacr.org/2011/074, 2011.

[BMT78] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, vol. 24, no.3, pp. 384–386, 1978.

[Cay08] P.-L. Cayrel, "Construction et optimisation de cryptosystèmes basés sur les codes correcteurs d'erreurs," *thèse de doctorat*, Université de Limoges, 2008.

[CCB11] N. Chikouche, F. Cherif, and M. Benmohammed, "Conception et Vérification d'un Protocole d'Authentification de Système Combiné RFID-Biométrique », *CEUR Workshop Proceedings, Conférence Internationale sur l'Informatique et ses Applications (CIIA'11)*, 2011.

[CCB12a] N. Chikouche, F. Cherif, and M. Benmohammed, "An Authentication Protocol Based on Combined RFID-Biometric System", *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 3, No.4, pp. 62-67, 2012.

[CCB12b] N. Chikouche, F. Cherif, and M. Benmohammed, "Vulnerabilities of two Recently RFID Authentication Protocols," *The IEEE International Conference on Complex Systems (ICCS'12)*, IEEE, 2012.

[CCB13] N. Chikouche, F. Cherif, and M. Benmohammed "Algebraic Replay Attacks on Authentication in RFID Protocols," *Advances in Security of Information and Communication Networks SecNet 2013*(A.I. Awad, et al., eds.) vol. CCIS 381, pp. 153–163. Springer, 2013.

[CCC+04] Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, J. Mantovani, S. Modersheim, and L. Vigneron, "A high level protocol specification language for industrial security-sensitive protocols," *In Proceedings of Workshop on Specification and Automated Processing of Security Requirements*, 2004.

[CCCB15a] N. Chikouche, F. Cherif, P-.L. Cayrel, and M. Benmohammed "Weaknesses in Two RFID Authentication Protocols," *Codes, Cryptography and Information Security C2SI 2015*, S. El Hajji et al., (eds.), vol. LNCS 9084, pp. 162–172, Springer, 2015.

[CCCB15b] N. Chikouche, F. Cherif, P-.L. Cayrel, and M. Benmohammed "Improved RFID Authentication Protocol Based on Randomized McEliece Cryptosystem,**"** *International Journal of Network Security*, vol. 17, no. 4, pp. 413–422, 2015.

[CCCB15c] N. Chikouche, F. Cherif, P-.L. Cayrel, and M. Benmohammed "A Secure Code-Based Authentication Scheme for RFID Systems,**"** *I.J. Computer Network and Information Security,* vol. 7, no. 9, pp. 1-9, 2015.

[CCZ+14] C. -M. Chen, S. -M. Chen, X. Zheng, L. Yan, H. Wang, and H.-M. Sun, "Pitfalls in an ECC-based lightweight authentication protocol for low-cost RFID," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 4, pp. 642-648, 2014.

[CDP09] X. Chen, T. van Deursen , and J. Pang, "Improving Automatic Verification of Security Protocols with XOR," i*n Proceedings of ICFEM 2009* (K. Breitman and A. Cavalcanti, eds.), vol. LNCS 5885, pp. 107 – 126, Springer, 2009.

[CFS01] N.T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," i*n Proceedings of ASIACRYPT 2001* (C. Boyd, eds.), vol. LNCS 2248, pp. 157–174, Springer, 2001.

[CH07] H.-Y. Chien and C.-W. Huang, "A lightweight RFID Protocol Using Substring," i*n: Embedded Ubiquitous Computing*, pp. 422–431, 2007.

[Chi06] H.-Y. Chien, "Secure access control schemes for RFID systems with anonymity," in *Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, 2006.

[Chi07] H.-Y. Chien, “SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 337-340, 2007.

[Chi13] H.-Y. Chien, “Combining Rabin cryptosystem and error correction codes to facilitate anonymous authentication with un-traceability for low-end devices,” *Computer Networks*, vol. 57, pp. 2705–2717, 2013.

[CKMI07] Y. Cui, K. Kobara, K. Matsuura, and H. Imai, “Lightweight asymmetric privacy-preserving authentication protocols secure against active attack,” in *Proceedings of IEEE PerComW’07*, 2007, pp. 223–228.

[CL09] H.-Y. Chien and C.-S. Laih, “ECC-based lightweight authentication protocol with untraceability for low-cost RFID,” *Journal of Parallel and Distributed Computing*, vol. 69, pp. 848–853, 2009.

[CS09] T. Cao, P. Shen, “Cryptanalysis of Two RFID Authentication Protocols,” *International Journal of Network Security*, vol. 9, N°1, pp. 95-100, 2009.

[CS98] A. Canteaut and N. Sendrier, “Cryptoanalysis of the original McEliece cryptosystem,” *In Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security : Advances in Cryptology*, vol. LNCS 1514, pp. 187–199, 1998.

[CVE10] P.-L. Cayrel, P. Véron, and S. M. El Yousfi Alaoui, “A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem,” In Selected Areas in Cryptography, vol. LNCS 6544, pp. 171–186, Springer, 2010.

[DH76] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no.6, pp. 644-654, November 1976.

[DMR08] T. van Deursen, S. Mauw, and S. Radomirović, “Untraceability of RFID protocols*,” Springer In Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, vol. 5019, pp.1-15, 2008.

[DOF05] S. Dominikus, E. Oswald, and M.Feldhofer, “Symmetric Authentication for RFID Systems in Practice,” *Handout of the Ecrypt Workshop on RFID and Lightweight Crypto*, 2005.

[Dom06] A. Dominguez, “Cryptanalysis of park’s authentication protocol in wireless mobile communication systems,” *International Journal of Network Security*, vol. 3, no. 3, p. 279-282, November 2006.

[DR08] T. van Deursen, S. Radomirovic, “Attacks on RFID Protocols,” *Report 2008/310*, *Cryptology ePrint Archive*, 2008.

[DR09] T. van Deursen and S. Radomirović, “Algebraic Attacks on RFID Protocols,” *in Proceedings of WISTP 2009* (O. Markowitch, et al., eds.), vol. LNCS 5746, pp. 38–51. Springer, 2009.

[DRL11] M. David, D. C. Ranasinghe, and T. Larsen, “A2U2: A Stream Cipher for Printed Electronics RFID tags,” *IEEE International Conference on RFID (RFID 2011)*, pp. 176-183, 2011.

[DSV03] L. Durante, R. Sisto, and A. Valenzano, “Automatic testing equivalence verification of Spi calculus specifications,” *ACM Trans. Softw. Eng.*

Methodol, vol. 12, no. 2, pp.222–284, 2003.

[DY83] D. Dolev and A. Yao, "On security of public key protocols," *IEEE transactions on Information Theory*, vol. 29, pp. 198–208, 1983.

[EL12] E. El Moustaine and M. Laurent,"A lattice based authentication for low-cost RFID", *International Conference IEEE on RFID-Technologies and Applications (RFID-TA)*, pp. 68-73, 2012.

[ElG85] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *Advances in Cryptology – CRYPTO '84*, vol. LNCS 196, pp. 10–18, 1985.

[Epc] EPC Global, http://www.gs1.org/epcglobal , 2015.

[Eva09] D. Evans, "Top 25 Technology Predictions," *CISCO Systems*, http://www.cisco.com, 2009.

[FDW04] M.Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," i*n Proceedings of CHES 2004*, vol. LNCS 3156, pp.357–370, Springer, 2004.

[Flo14] R. Floyd, "RFID in Transportation", www.enginnering.com, 2014,

[FS86] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," *In CRYPTO '86*, vol. LNCS 263, pp. 186–194, 1986.

[FS96] J.-B. Fischer and J. Stern, "An efficient pseudo-random generator provably as secure as syndrome decoding," in *Proceedings of EUROCRYPT 1996* (U.M. *Maurer, eds.),* vol. LNCS 1070, pp. 245–255, Springer, 1996.

[FS09] M. Finiasz, and N. Sendrier, "Security Bounds for the Design of Code-based Cryptosystems," *In Advances in Cryptology – Asiacrypt 2009*, vol. LNCS 5912, pp 88-105, 2009.

[GGH97] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptosystems from lattice reduction problems," in *Advances in cryptology*, vol. LNCS 1294, pp. 112–131, 1997.

[GGHS09] Y. Glouche, T. Genet, O. Heen, E. Houssay and R. Saillard, "SPAN (a Security Protocol ANimator for AVISPA)," http://www.irisa.fr/celtique/genet/span/, 2009.

[GK00] T. Genet and F. Klay. "Rewriting for cryptographic protocol verification," *In: 17th International Conference on Automated Deduction* (D. McAllester, eds.), vol. LNCS 1831, pp.271–290, Springer, 2000.

[GM82] S. Goldwasser and S. Micali, "Probabilistic encryption and how to play mental poker keeping secret all partial information," in *Proceedings of STOC*, p.p. 365–377, 1982.

[Gop70] V. D. Goppa, "A new class of linear correcting codes," *Probl. Peredachi Inf.*, vol. 6, no. 3, pp. 24-30, 1970.

[GPP11] J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON Family of Lightweight Hash Functions," *31st Annual Cryptology Conference*, pp.222-239, 2011.

[GTT 03] T. Genet, Y. Tang-Talpin -M., V.V.T. Tong, "Verification of copy-protection cryptographic protocol using approximations of term rewriting systems," in *Proc. of the Workshop on Issues in the Theory of Security (WITS'03)*, 2003.

[Hal10] J.I.Hall, “Notes on Coding Theory,” www.math.msu.edu\~jhall, 2010.

[Ham50] R. W. Hamming, “Error detecting and error correcting codes,” *Bell System Technical Journal*, vol. 29, no.2, pp. 147–160, 1950.

[Han06] G.P. Hancke, “Practical Attacks on Proximity Identification Systems,” in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 328–333, 2006.

[Hei87] R. Heiman, “On the security of cryptosystems based on linear error-correcting codes,” *Master’s thesis*, Feinberg Graduate School of the Weizman Institute of Science, 1987.

[HKCL14] D. He, N. Kumar, N. Chilamkurti, and J.-H. Lee, “Lightweight ECC Based RFID Authentication Integrated with an ID Verifier Transfer Protocol,” *Journal of Medical Systems*, vol. 38, pp. 1-6, 2014.

[HMG13] S. Heyse, I. von Maurich, and T. Güneysu, “Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices,” in *Proceedings of CHES 2013* (G. Bertoni and J.-S. Coron, Eds.), 2013.

[HPS98] J. Hoffstein, J. Pipher, J.H. Silverman, “A ring based public key Cryptosystem,” in *Proceedings of ANTS-III*, vol. LNCS 1423, pp. 267–288, Springer, 1998.

[IK01] H. Imai and K. Kobara, “Semantically secure mceliece public-key cryptosystems – conversions for McEliece PKC,” in *Proceedings of 4th International Workshop on Practice and Theory in Public Key Cryptosystems (PKC 01)*, pp. 19–35, 2001.

[JDTL12] H. Jialiang, O. Dantong, B. Tian, Z. Liming, "A Lightweight RFID Authentication Protocol for Mobile Reader", *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 6, pp. 80-88, 2012.

[JF12] H. Jannati, A. Falahati, “Cryptanalysis and Enhanced of Two Low Cost RFID Authentication protocols,” *International Journal of UbiComp*, vol. 3, no. 1, pp. 1-9, 2012.

[JW07] A. Juels and S.A. Weis, “Defining strong privacy for RFID,” in *Proceedings of PerCom’ 07*, http://eprint.iacr.org/2005/049, 2007.

[KGA15] A. Kumar, K. Gopal, and A. Alok, "A Novel Trusted Hierarchy Construction for RFID-Sensor Based MANETs Using ECC," *ETRI Journal*, vol. 37, no. 1, pp. 186-196 , 2015

[Khe14] W. Khedr, “On the Security of Moessner’s and Khan’s Authentication Scheme for Passive EPCglobal C1G2 RFID Tags,” *International Journal of Network Security*, vol. 16, no. 5, pp. 369–375, 2014.

[KI06] K. Kobara and H. Imai, “Personalized-public-key cryptosystem (P2KC)-application where public-key size of Niederreiter PKC can be reduced,” in *Workshop on Codes and Lattices in Cryptography (CLC2006)*, pp. 61–68, 2006.

[KKK97] G. Kabatianskii, E. Krouk, and B.J.M. Smeets, “A digital signature scheme based on random error-correcting codes,” *Cryptography and Coding* (M.J. Darnell, eds.), vol. LNCS 1355, pp. 161–167. Springer, 1997.

[KW96] D. Kindred, J. Wing, “Fast, automatic checking of security protocols,” in *Proceedings of USENIX 2nd Workshop on Electronic Commerce*, pp. 41–52, 1996.

[KZW08] M.K. Khan, J. Zhang, and X. Wang, "Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices," *Chaos, Solitons and Fractals*, vol. 35, no. 3, pp. 519-524, 2008.

[LAK06] S. Lee, T. Asano, and K. Kim, "RFID mutual authentication scheme based on synchronized secret information," in *Symposium on Cryptography and Information Security*, 2006.

[LCC10] Y.W. Lai, S.-C. Chang, and C. Chang, "An Improved Biometrics-based User Authentication Scheme without Concurrency System," *International Journal of Intelligent Information Processing,* vol. 1, no. 1, pp. 41-49, 2010.

[LH10] C.T. Li and M.S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1-5, 2010.

[LHLL05] S.M. Lee, Y.J. Hwang, D.H. Lee, and J.I. Lim. "Efficient Authentication for Low-Cost RFID Systems," *International Conference on Computational Science and its Applications - ICCSA 2005*, 2005.

[LHYC08] Y.C. Lee, Y.C. Hsieh, P.S. You, and T.C. Chen, "An Improvement on RFID Authentication Protocol with Privacy Protection," *Third International Conferences on Convergence and Hybrid Information Technology*, vol. 2, pp. 569–573, 2008.

[Liu08] Y. Liu, " An Efficient RFID Authentication Protocol for Low-Cost Tags," in *Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, pp. 706–711, ,2008.

[LOK08] J. Lim, H. Oh, S. Kim, "A New Hash-Based RFID Mutual Authentication Protocol Providing Enhanced User Privacy Protection," in *Proceedings of ISPEC 2008*, vol. LNCS 4991, Springer, pp. 278–289, 2008.

[LYL14] Z. Li, R. Zhang, Y. Yang, and Z. Li, "A provable secure mutual RFID authentication protocol based on error-correct code," in *Proceedings of 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. IEEE, pp. 73–78, 2014.

[MB09] R. Misoczki and P. S. L. M. Barreto, "Compact McEliece keys from Goppa codes," in *Selected Areas in Cryptography 2009 (SAC 2009)*, vol. LNCS 5867, pp. 376–392, 2009.

[McE78] R. J. McEliece. "A public-key system based on algebraic coding theory," Tech. Rep. DSN Progress Report 44, Jet Propulsion Lab, 1978.

[MG14] I. von Maurich and T. Güneysu, "Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices," in *Proceedings of the Conference on Design, Automation & Test in Europe, DATE '14*, pp. 1–6, IEEE, 2014.

[Mif] *The Mifar cards*, http://www.mifare.net, 2015.

[Mih11] M.I. Mihailescu, "Resreach on Solutions for Preventing Algebraic Attacks Against Biometric and RFID Protocols," *ACTA Universitatis Apulensis (Special Issue)*, pp. 371- 386, 2011.

[Mil85] V. S. Miller, "Use of Elliptic Curves in Cryptography", Advances in cryptology – CRYPTO'85, vol. LNCS 218, pp. 417-426, 1985.

[MM12] B. Malek and A. Miri, "Lightweight mutual RFID authentication," in *Proceedings of IEEE International Conference on Communications*. pp. 868–872, IEEE, 2012.

[MRS86] S. Micali, C. Rackoff, and B. Sloan, "The notion of security for probabilistic cryptosystems," In *Proceedings of CRYPTO 86*, pp. 381–392, 1986

[MRT08] A. Mitrokotsa, M.R. Rieback, and A.S. Tanenbaum "Classification of RFID Attacks," *International Workshop on RFID Technology (IWRT)*, pp. 73-86, 2008.

[MS77] F.J. MacWilliams and N.J.A. Sloane,"The Theory of Error-Correcting Code," Wiley, 1982.

[MTSB13] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes," *ISIT 2013*, pp. 2069-2073, 2013.

[Nie86] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Problems Control Inform. Theory*, vol. 15, no. 2, pp. 159–166, 1986.

[NIKM08] R. Nojima, H. Imai, K. Kobara, and K. Morozov, "Semantic security for the McEliece cryptosystem without random oracles," *Designs, Codes and Cryptography*, vol. 49, no. 1–3, pp. 289–305, 2008.

[NNJ10] A. Nagar, K. Nandakumar, and A.K. Jain, "Biometric template transformation: a security analysis", in *Proc. Media Forensics and Security*, 2010.

[OP08] K. Ouafi and R.C.-W Phan, "Privacy of recent RFID authentication protocols," in *Proceedings of ISPEC 2008*, LNCS, vol. 4991, pp. 263–277. Springer, 2008.

[Oua12] K. Ouafi, "Security and Privacy in RFID Systems," Phd thesis, École polytechnique fédérale de Lausanne, 2012.

[Park04] S. Park, "Authentication protocol providing user anonymity and untraceability in wireless mobile communication systems," *Computer Networks*, vol. 44, pp. 267–273, 2004.

[PCMA06] P.-L. Pedro, H.-C. Julio Cesar, E.-T. Juan M., R. Arturo, "EMAP: An Efficient Mutual Authentication Protocol for Low-cost RFID Tags", *OTM Federated Conferences and Workshop: IS Workshop -- IS'06,* LNCS, vol. 4277, P-352--361, 2006.

[QYY08] C. Qingling, Z. Yiju, W. Yonghua, "A minimalist mutual authentication protocol for RFID system & BAN logic analysis," *In: Proc. of CCCM 2008*, pp. 449–453. IEEE Computer Society, Los Alamitos, 2008.

[RC14] M. Repka and P.-L. Cayrel, "Cryptography Based on Error Correcting Codes : A Survey," *Multidisc. Persp. In Cryptology and Information Security*, pp. 133-156, 2014.

[RHV09] Joel J.P.C. Rodrigues, F.D. Heirto, and B. Vaidya "Decentralized RFID authentication Solution for embedded Systems," *4th Int. Conference on Systems and Networks Communications*, IEEE, pp. 174-178, 2009.

[RSA78] R.L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, N°2, pp. 120–126, 1978.

[SBN11] M. Safkhani, N. Bagheri, and M. Naderi, "Cryptanalysis of Chen et al.'s RFID Access Control Protocol," In: IACR Cryptology ePrint Archive, vol. 2011, pp.194, 2011.

[Sch72] P.M Schalkwijk, "An algorithm for source coding," *IEEE Transactions of Information theory*, vol. 18, no. 3, pp. 395–399, 1972.

[Sen05] N. Sendrier, "Encoding information into constant weight words," in *IEEE conference, ISIT'2005*, pp. 435–438, 2005.

[Sho94] Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 20–22, 1994.

[SKI06] M. Suzuki, K. Kobara, and H. Imai, "Privacy enhanced and light weight RFID system without tag synchronization and exhaustive search," in *Proceedings of 2006 IEEE International Conference on Systems, Man, and Cybernetics*, 2006, pp. 1250–1255.

[SKI10] T. Sekino, Y. Cui, K. Kobara, and H. Imai, "Privacy enhanced RFID using Quasi-Dyadic fix domain shrinking," in *Proceedings of Global Telecommunications Conference*, pp. 1–5, IEEE, 2010.

[Sma11] SmartCard Alliance. "Smart Cards and Biometrics," available at: wwww.smartcardalliance.org , 2011.

[Son99] D. X. Song, "Athena: A New Efficient Automatic Checker for Security Protocol Analysis," Proc. of the 12th Computer Security Foundations Workshop (CSFW'99), IEEE Computer Society, June 1999.

[SSM05] Y. Sutcu, H-T. Sencar, and N. Memon. "A Secure Biometric Authentication Scheme Based on Robust Hashing," MM-SEC'05, New York, USA, August 1–2, 2005.

[TCL12] Y. Tian, G. Chen, and J. Li, "A new ultralightweight RFID authentication protocol with permutation," *IEEE Communications Letters*, vol. 16, pp. 702–705, 2012.

[Tea06] AVISPA team, "HLPSL Tutorial The Beginner's Guide to Modelling and Analysing Internet Security Protocols," Technical report, AVISPA project, 2006.

[Vér96] P. Véron, "Improved Identification Schemes Based on Error-Correcting Codes," *Appl. Algebra Eng. Commun. Comput.*, vol. 8, no.1, pp.57–69, 1996.

[XPK14] H. Xin,Y. Pin,L. Kun, "NTRU-based RFID tripartite authentication protocol," *Computer Engineering Applications*, vol. 50, no. 3, pp. 63-66, 2014.

[WHC11] C.-H Wei, M.-S Hwang, A.-Y Chin, "A Mutual Authentication Protocol for RFID", *IT Professional*, IEEE Computer Society, vol.3, pp.20-24, 2011.

[WSRE03] S. Weis, S. Sarma, R. Rivest, and D. Engels. "Security and privacy aspects of low-cost radio frequency identification systems," In D. Hutter, and all., editors, *International Conference on Security in Pervasive Computing – SPC 2003*, vol. LNCS 2802, pp.454–469, Springer, 2003.

[Zen09] E. Zenner, "Authentication for RFID Tags: Observations on the HB Protocols," *4th Interdisciplinary Seminar on Applied Mathematics*, pp. 1-20, 2009.

[ZZC14] X. Zhuang, Y. Zhu, and C-C. Chang, “A New Ultralightweight RFID Protocol for Low-Cost Tags: R2AP,” *Wireless Pers Commun*, vol. 79, no. 3, pp. 1787–1802, 2014,

Appendix

Appendix A: *HLPSL of Wei et al protocol*

```
role reader ( R,T: agent, ID,RID, S: text, H : hash_func, Snd,Rec:
channel(dy))
    played_by R
    def=
    local  State         : nat,
    Nr, Nt, Ndb            : text
    init State := 0
    transition
    1. State = 0  /\ Rec(start)   =|> State' := 1 /\ Nr' := new()  /\
Snd(Nr')
    2. State = 1  /\ Rec(H(xor(xor(S,Nr),Nt')).Nt')
    =|> State' := 2   /\ Ndb' := new() /\ Snd(H(xor(ID,Ndb')).Ndb') /\
secret(ID,sec_id,{R,T})
    /\ request(R,T,aut_tag,Nt') /\ witness(R,T,aut_reader,Ndb')
end role

role tag ( T,R: agent, ID,RID,S: text, H : hash_func,Snd,Rec:
channel(dy))
    played_by T
    def=
    local  State               : nat,
    Nt, Nr,Ndb                  : text
 %const sec_k2 : protocol_id
    init State := 0
    transition
    1. State = 0 /\ Rec(Nr')   =|>   State' := 1   /\ Nt' := new()
    /\ Snd(H(xor(xor(S,Nr'),Nt')).Nt') /\ witness(T,R,aut_tag,Nt')

    2. State = 1 /\ Rec(H(xor(ID,Ndb')).Ndb')
    =|> State' := 2 /\ request(T,R,aut_reader,Ndb')
end role
role session(R,T : agent,ID,RID,S : text, H: hash_func)
def=
  local Sa,Ra,Sb,Rb : channel(dy)
    composition
    reader(R,T,ID,RID,S,H,Sa,Ra) /\  tag(T,R,ID,RID,S,H,Sb,Rb)
end role
role environment() def=
const r,t : agent,
      id,rid,s,id1,s1: text,
```

```
      h: hash_func,
      aut_reader, aut_tag, sec_id : protocol_id
      intruder_knowledge = {r,t,h}
    composition
    session(r,t,id,rid,s,h)
    /\  session(r,t,id,s,h)
end role
goal
 secrecy_of sec_id
authentication_on aut_tag
authentication_on aut_reader
end goal
environment()
```

Appendix B: *HLPSL of Jialiang et al. protocol*

```
role reader ( R,T: agent, ID,S: text, H : hash_func, Snd,Rec:
channel(dy))
    played_by R
    def=
    local  State         : nat,
    Nr, Nt, Ndb            : text
    const sec_id : protocol_id
    init State := 0
    transition
    1. State = 0  /\ Rec(start)   =|> State' := 1 /\ Nr' := new()  /\
Snd(Nr')
    2. State = 1  /\ Rec(Nt'.xor(S,H(xor(Nr,Nt'))))
    =|> State' := 2   /\ Ndb' := new() /\
Snd(xor(H(xor(xor(Nr,Nt'),Ndb')),ID)) /\ secret(ID,sec_id,{R,T})
    /\ request(R,T,aut_tag,Nt') /\ witness(R,T,aut_reader,Ndb')
end role

role tag ( T,R: agent, ID,S: text, H : hash_func,Snd,Rec: channel(dy))
    played_by T
    def=
    local  State               : nat,
    Nt, Nr,Ndb                  : text
 %const sec_k2 : protocol_id
    init State := 0
    transition
    1. State = 0 /\ Rec(Nr')   =|>   State' := 1   /\ Nt' := new()
    /\ Snd(Nt'.xor(S,H(xor(Nr',Nt')))) /\ witness(T,R,aut_tag,Nt')

    2. State = 1 /\ Rec(xor(H(xor(xor(Nr,Nt),Ndb')),ID))
    =|> State' := 2 /\ request(T,R,aut_reader,Ndb')
end role

role session(R,T : agent,ID,S : text, H: hash_func)
def=
  local Sa,Ra,Sb,Rb : channel(dy)
    composition
    reader(R,T,ID,S,H,Sa,Ra) /\  tag(T,R,ID,S,H,Sb,Rb)
end role

role environment() def=
```

```
const r,t : agent,
      id,s,id1,s1: text,
      h: hash_func,
      aut_reader, aut_tag : protocol_id
      intruder_knowledge = {r,t,h}
    composition
    session(r,t,id,s,h)
    /\  session(r,t1,id1,s1,h)
    /\  session(r,t,id,s,h)
end role

goal
 secrecy_of sec_id
authentication_on aut_tag
authentication_on aut_reader

end goal

environment()
```

Appendix C: *HLPSL of our RFID-Biometric Authentication protocol*

```
role reader ( R,T: agent, ID,B : text, H,G,Hright,Hleft : hash_func,
              Snd,Rec: channel(dy))
    played_by R
    def=
      local  State  : nat, Nr, Nt : text, HB: message
      const sec_id1 : protocol_id
      init State := 0
      transition
       1. State = 0 /\ Rec(start)  =|>  State' := 1
          /\ Nr' := new() /\ Snd(Nr')
          /\ witness(R,T,aut_reader,Nr')
       2. State = 1
          /\ Rec( Nt'.Hleft(Nt',xor(ID,Nt'),Nr))
          =|>  State' := 2 /\ Snd(Hright(Nt',xor(ID,Nt'),Nr))
          /\ request(R,T,aut_tag,Nt') /\ secret(ID,sec_id1,{R,T})
3. State=2
     /\ Rec( xor(H(ID,Nt,Nr),HB))  =|>  State' := 3
end role
    role tag ( T,R: agent, ID : text, HB: message,
                H,G,Hright,Hleft : hash_func,
               Snd,Rec: channel(dy))
    played_by T
    def=
      local  State  : nat, Nt, Nr : text, B: text
      const sec_id2, sec_b: protocol_id
      init State := 0
      transition
       1. State = 0 /\ Rec(Nr') =|> State' := 1 /\ Nt' := new()
        /\ Snd( Nt'.Hleft(Nt',xor(ID,Nt'),Nr'))
```

```
        /\ witness(T,R,aut_tag,Nt') /\ secret(ID,sec_id2,{T,R})
       2. State = 1 /\ Rec(Hright(Nt,xor(ID,Nr),Nt)) =|>
          State' := 2 /\ request(T,R,aut_reader,Nr)
           /\ Snd( xor(H(ID,Nt,Nr),HB))  /\ secret(HB,sec_b,{T,R})
end role
role session(T,R : agent, ID,B : text, H,G,Hright,Hleft : hash_func)
def=
local Se,Re,Sf,Rf : channel(dy)
const aut_reader, aut_tag : protocol_id
composition
tag(T,R,ID,G(B),H,G,Hright,Hleft,Se,Re)
/\ reader(R,T,ID,B,H,G,Hright,Hleft,Sf,Rf)
end role
role environment() def=
const t,r : agent,
      id,b,idti,idri,bti,bri : text,
      h,g,hleft,hright : hash_func
intruder_knowledge = {t,r,h,g,hleft,hright,idti,idri,bti,bri}
composition
session(t,r,id,b,h,g,hright,hleft)
/\session(t,r,id,b,h,g,hright,hleft)
 /\  session(t,i,idti,bti,h,g,hright,hleft)
/\ session(i,r,idri,bri,h,g,hright,hleft)
end role
    goal
    secrecy_of  sec_id1, sec_id2 ,sec_b
    authentication_on aut_reader
    authentication_on aut_tag
    end goal
    environment()
```

Appendix D: ***HLPSL of our improved protocol based on randomized McEliece cryptosystem (R2McE)***

```
role reader ( R,T: agent, ID,Rold, Rnew: text,
              Fg,Phi : hash_func, KG: public_key,
              Snd,Rec: channel(dy))
    played_by R
    def=
      local  State  : nat,
             Nr, X, RN : text, E : hash(text),
              DID,DNew : {text.text}_public_key
      init State := 0
      transition
       1. State = 0
          /\ Rec(start)  =|>  State' := 1 /\ Nr' := new()
          /\ Snd(Nr') /\ witness(R,T,aut_reader,Nr')
        % if CR= CRnew
       2. State = 1
          /\ Rec({DID}_E'.Fg(Nr.X'.DID)) =|> State' := 2
          /\ RN':=new()  /\ DNew':={ID.RN'}_KG
          /\ Snd(xor(DNew',E').Fg(Nr.DNew'.X')) /\
secret({DNew'},sec_did2, {R,T})
          /\ request(R,T,aut_tag,X') /\ Rold':=Rnew /\ Rnew':=RN'
       % if CR= CRold
         3. State = 1
          /\ Rec({DID}_E'.Fg(Nr.X'.DID)) =|> State' := 2
           /\ DNew':={ID.Rnew}_KG
          /\ Snd(xor(DNew',E').Fg(Nr.DNew'.X')) /\
secret({DNew'},sec_did2, {R,T}) /\ request(R,T,aut_tag,X')
end role
role tag ( T,R: agent, DID: {text.text}_public_key,
            Fg,Phi : hash_func,
            Snd,Rec: channel(dy))
    played_by T
    def=
      local  State  : nat,
             Nr, X, RN : text,
             E: hash(text), DNew: {text.text}_public_key
    init State := 0
    transition
     1. State = 0 /\ Rec(Nr') =|> State' := 1
        /\ X' := new()  /\ E':=Phi(X')
```

```
        /\ Snd({DID}_E'.Fg(Nr'.X'.DID)) /\ witness(T,R,aut_tag,X')
        /\ secret({DID},sec_did1, {T,R})
     2. State = 1 /\ Rec(xor(DNew',E).Fg(Nr.DNew'.X'))
          =|> State' := 2
        /\ request(T,R,aut_reader,Nr) /\ DID' := DNew'
end role
role session(R,T: agent, ID,Rinit: text,
             Fg, Phi : hash_func, KG: public_key)
def=
local Se,Re,Sf,Rf : channel(dy)
const aut_reader, aut_tag, sec_did1, sec_did2 : protocol_id
composition
tag(T,R,{ID.Rinit}_KG,Fg,Phi,Se,Re)
/\ reader(R,T,ID,Rinit,Rinit,Fg,Phi,KG, Sf,Rf)
end role
role environment() def=
const t,r,i : agent, id,rinit,idit,idri: text,
      g,phi : hash_func, kG,kGti,kGri: public_key
 intruder_knowledge = {t,r,i,g,kG,phi,kGti,kGri,idit,idri}
composition
% To detection a replay attack:
session(r,t,id,rinit,g,phi,kG)
/\ session(r,t,id,rinit,g,phi,kG)
/\ session(i,t,idit,rinit,g,phi,kGti)
/\ session(r,i,idri,rinit,g,phi,kGri)
end role
    goal
       secrecy_of sec_did1 % confidentiality of DID
       secrecy_of sec_did2 % confidentiality of DNew
       authentication_on aut_reader % Reader's authentication
     authentication_on aut_tag % Tag's authentication
    end goal
    environment()
```

Appendix E: *HLPSL of our improved protocol based on QC-MDPC McEliece cryptosystem (RQMcE)*

```
role tag (T,R: agent, ID,Rand: text,
          Fg,Right : hash_func,
         PKG: public_key,
         Snd,Rec: channel(dy))
    played_by T
    def=
      local  State  : nat,
             Nr, E, Randp : text
    init State := 0
    transition
1. State = 0 /\ Rec(Nr') =|> State' := 1
  /\ E' := new()
  /\ Snd({{ID.Rand}_PKG}_E'.Fg(ID.Nr'.E'))
  /\ witness(T,R,tag_auth,E')
  /\ secret({ID},sec_id, {T,R})
  /\ secret({Rand},sec_rand, {T,R})
2. State = 1 /\
 Rec(xor(Randp',Right(E)).Fg(ID.Nr.Randp'))
   =|> State' := 2
   /\ request(T,R,reader_auth,Nr)
   /\  Rand':=Randp'
end role

role reader ( R,T: agent,
              ID,Rnew,Rold: text,
              Fg,Right : hash_func,
             PKG: public_key,
              Snd,Rec: channel(dy))
    played_by R
    def=
      local  State  : nat,
             Nr, E, Randp : text
      init State := 0
      transition
1. State = 0 /\ Rec(start)  =|>
   State' := 1 /\ Nr' := new() /\ Snd(Nr')
   /\ witness(R,T,reader_auth,Nr')

2. State = 1 /\
```

```
   Rec({{ID.Rnew}_PKG}_E'.Fg(ID.Nr.E'))
   =|> State' := 2 /\ Randp':= new()
   /\ request(R,T,tag_auth,E') /\
Snd(xor(Randp',Right(E')).Fg(ID.Nr.Randp'))
   /\  Rold':=Rnew /\ Rnew':=Randp'
   /\ secret({Randp'},sec_randp, {R,T})

2. State = 1 /\
    Rec({{ID.Rold}_PKG}_E'.Fg(ID.Nr.E'))
    =|> State' := 2 /\ Randp':= Rnew
    /\ request(R,T,tag_auth,E') /\
Snd(xor(Randp',Right(E')).Fg(ID.Nr.Randp'))
    /\ secret({Randp'},sec_randp, {R,T})
end role

role session(R,T: agent, ID,Rand: text,
             Fg,Right : hash_func,
             PKG: public_key)
  def=
  local Se,Re,Sf,Rf : channel(dy)
  const reader_auth, tag_auth, sec_id,
  sec_rand,sec_randp : protocol_id
  composition
  tag(T,R,ID,Rand,Fg,Right,PKG, Se,Re)
  /\ reader(R,T,ID,Rand,Rand,Fg,Right,PKG,
     Sf,Rf)
end role

role environment() def=
  const t,r,i : agent, id,rand: text,
      g,right : hash_func,
      pkG: public_key

  intruder_knowledge = {t,r,i,g,right,pkG}
  composition
     session(r,t,id,rand,g,right,pkG)
  /\ session(r,t,id,rand,g,right,pkG)

end role

goal
```

```
   secrecy_of sec_id
   secrecy_of sec_rand
   secrecy_of sec_randp
   authentication_on reader_auth
   authentication_on tag_auth
end goal

    environment()
```