Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de
Docteur de Télécom & Management SudParis

Doctorat conjoint
Telecom & Management SudParis (TMSP) – Université Pierre et Marie Curie (UPMC)

Spécialité : informatique et réseaux

Par

Mohamed ABID

Des mécanismes d'authentification basés sur l'identité de l'utilisateur pour renforcer la sécurité des réseaux

Soutenue le 01/02/2011 devant le jury composé de :

Hatem BETTAHAR	Rapporteur	Université de Technologie de Compiègne
Lionel BRUNIE	Rapporteur	INSA Lyon
Bernadette DORIZZI	Examinatrice	Institut Telecom SudParis
Guy PUJOLLE	Examinateur	UPMC Paris 6
Hassnaa MOUSTAFA	Examinatrice	Telecom R&D (Orange Labs)
Patrick SENAC	Examinateur	ISAE Toulouse
Dijana PETROVSKA-DELACRÉTAZ	Examinatrice	Institut Telecom SudParis
Hossam AFIFI	Directeur de thèse	Institut Telecom SudParis

Thèse n° 2011TELE0005

Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de
Docteur de Télécom & Management SudParis

Doctorat conjoint
Telecom & Management SudParis (TMSP) – Université Pierre et Marie Curie (UPMC)

Spécialité : informatique et réseaux

Par
Mohamed ABID

User Identity Based Authentication Mechanisms for Network Security Enhancement

Soutenue le 01/02/2011 devant le jury composé de :

Hatem BETTAHAR	Rapporteur	Université de Technologie de Compiègne
Lionel BRUNIE	Rapporteur	INSA Lyon
Bernadette DORIZZI	Examinatrice	Institut Telecom SudParis
Guy PUJOLLE	Examinateur	UPMC Paris 6
Hassnaa MOUSTAFA	Examinatrice	Telecom R&D (Orange Labs)
Patrick SENAC	Examinateur	ISAE Toulouse
Dijana PETROVSKA-DELACRÉTAZ	Examinatrice	Institut Telecom SudParis
Hossam AFIFI	Directeur de thèse	Institut Telecom SudParis

Thèse n° 2011TELE0005

Abstract

In this thesis, we design new authentication mechanisms based on user identity. Therefore, we bring improvements in access control for different classes of networks such as Home Network, Governmental Network and Cellular Network. The identity can be biometric public features, simple strings (email addresses, login...), etc. The goal of our work is to design innovative solutions that secure and personalize authentication mechanisms. We have three main solutions in the thesis depending on the deployed identity.

The first solution concerns the use of biometric in Home Network' authentication mechanisms. In the Home Network (HN) case study, we aim at personalizing the access of each user in the HN and preventing illegitimate users (passing by the HG) to have any access. Our approach of personalized access also permits each user to use any device in the HN, while being able to access his/her appropriate profile. We propose a new biometric authentication method, while keeping in mind the constraint of the non storage of the users' Biometric Template BT in the Home Gateway (HG). To satisfy this constraint, we propose using the *fuzzy vault* method to hide a secret that should be used for authentication. The HG has the role of generating a secret for each user session, which are hidden by the BT. The user needs to recover the secret in order to be authenticated. The personalized users' access in the proposed solution, allows controlling the access for each broadband access line depending on the user that is being connected.

The second solution proposes e-Passport authentication mechanisms. The cryptographic parameters are generated using the biometric templates and hence, personalized for the user. In travel document case study, we present our proposal which introduces a new e-Passport authentication mechanisms based on the *Elliptic Curve Diffie-Hellman* (ECDH) Key Agreement protocol. This protocol is needed to generate a session key used to authenticate the traveler and the Inspection System (IS) to exchange secure data. We designed two protocols. In the first one, the elliptic curve, used in the biometric cryptosystem, are generated from the minutiae data (fingerprint) of the e-Passport holder. In the second one, we use iris code to generate the elliptic curve. We analyzed the security of our solution with respect to the goals that we defined. We found that our solution fulfills its goals and prevents the system from the attacks. The use of biometric in the cryptographic solution is a very important issue as this biometric data is stored in the e-Passport Chip without a direct link to the security. This solution is validated by using iris biometrics. We performed tests on the NIST-ICE database of iris images to compute the False Rejection Rate and the False Acceptance Rate. The results obtained (e.g., FRR of 0.2% and FAR of 3.6%) are satisfying and the use of iris biometrics is encouraging for the deploying of this solution.

In the third solution, we worked on the Cellular Network and we used a simple string, like email addresses, as identifier to access to services. We choose the IP Multimedia Subsystem (IMS) which is an overlay architecture for the provision of multimedia services. We design a new service authentication mechanism relying on Identity Based Cryptography (IBC) for the IMS architecture. The goal was to authenticate the users using their public and private identifiers to overcome known weaknesses in the Authentication and Key Agreement (AKA) protocol. Security is assured using a symmetric protocol with a shared key (ks) between the User Equipement (UE) and the Home Subscriber System (HSS), an asymmetric protocol for signature, and Diffie-Hellman for key agreement. We focused on the eavesdropping and impersonation attacks that can take place in classical IMS scenario and we showed how our proposed solution can prevent against these attacks. We, then, proposed to add a Batch Verification on the Bootstrapping Server Function (BSF) to decrease signature verification delay and the authentication response time. To validate the performance of our proposed solution, we implemented the cryptographic operation in our proposed solution including the IBC procedures. We observed that the use of asymmetric cryptographic procedures leads to longer running time than symmetric procedures. However, the Batch Verification helps the BSF to verify the User Equipments (UEs) signature in a reasonable time.

Key words : Authentication, Biometrics, Identity Based Cryptography (IBC), IP Multimedia Subsystem (IMS), Home Network (HN)

Dedication

For my Father's memory,
For my Grand parents memory,
For my wife Najet,
For my son Ali,
For my mother,
*For **Prof. Hatem Bettaher** memory*
For my sisters and brother,
For my -in laws (father, mother and brothers),
For all those I appreciate...

Acknowledgements

In this thesis, I have been supported and supervised by many people to whom I would like to express my deepest gratitude:

- My supervisor, **Prof. Hossam Afifi**: thank you for the stupendous effort you spent to be a great colleague and friend.

- **Prof. Hatem Bettaher** and **Prof. Lionel Brunie** for accepting to review my dissertation and to be member of my Phd defense jury.

- **Prof. Bernadette Dorizzi**, **Prof. Guy Pujolle**, **Dr. Hassnaa Moustafa**, **Prof. Patrick Senac**, **Dr. Dijana Petrovska-Delacretaz** to be member of my Phd defense jury.

- The staff of TMSP; particularly, I want to thank: **Dr. Abdallah M'hamed**, **Dr. Vincent Gauthier**, . . . and especially our head of department **Prof. Djamal Zeghlache** for his helpful suggestions and advices. A very big thank goes to **Isabelle Rebillard** and **Valerie Mateus** for their patience and support.

- A special thank for **Dr. Hassnaa Moustafa** and **Dr. Dijana Petrovska-Delacretaz** for their help and valuable advices.

- A special thank for **Dr. Eric Renault**, **Shoaib Saleem** and **Mehdi Mani** for helping me to improve the language in my report.

- This thesis reports implementation results that were conducted with the help of several colleagues. A special thank goes to **Sondes Bannouri** and **Dingqi Yang** for their contribution to the two demonstrators.

- A special thank for **Dorsaf Zekri** and **Amira Bradai** for helping me to defend my thesis.

- My family, especially my wife, my son, my mother, my systers and brother, my -in laws (father, mother and brothers) for the support, help, and encouragement.

- My friends, both from TMSP and those who go back longer, especially **Boutabia**, **Emad**, **Chedhly**, **Aroua**, **Teck**, **Bastien**, **Ahmad**, **khaled**, **Ghazi**, **Sanjey Kanade** and **Songbo Song**.

This thesis was supported by a research studentship from the ANR/RNRT France (Agence Nationale de la recherche/ Réseau National de Recherche en Télécommunications). Project **BIOTYFUL**: BIOmetrics and crypTographY for Fair aUthentication Licensing [ANR-06-TCOM-018] (2006).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Network Security

Network security was and will stay a major catalyst for research in computer science. It begins with user authentication, commonly with a username and a password.

Security Management depends on the network classes. A small home or an agency needs only basic security, however, large businesses such as cellular operator networks require high network security management, more powerful software and hardware to avoid malicious attacks.

In this dissertation, we use different kind of identities to improve access control in such different networks, like Home Network (HN) scenario, e-Passport scenario and in final, the IP Multimedia Subsystem (IMS) scenario.

In this section, we first explore the concepts of identity, authentication, and authorization to clarify the differences between them. Then, we highlight the problem statement and our contribution. In the end, we present the organization of the thesis.

1.1.1 Identity

To access a system, users declare their identities to the system. This is the first response by the users to the system query "Who are you". Some familiar examples of identity are user IDs, digital certificates (which contain public keys) and credit cards. An important characteristic of identity is that it is public, and it is defined by: *identity is the affirmation that you are the legitimate owner of it and you defend this statement by using something accessible to everyone.*

1.1.2 Authentication

This is the answer of the system's query "Fine, but how can you prove it?". Generally, people consider password as the only mean for authentication. Although the passwords are the most common authenticators but there are in fact other authentication mechanisms. These mechanisms rely on the verification of one or more of the following:

- *Something you know*: This refers to the password authentication mechanism. As password are something that we know, we can forget it as well.
- *Something you have*: This is the answer to the forgetting problem. Users need to have something with them like smart card, Radio Frequency Identification (RFID) card, etc... But, these cards can be stolen.
- *Something you are*: In this case, the mechanism is called biometric authentication scheme. This scheme uses techniques such as fingerprints, retina scans, voice print analysis, etc... In computer science, biometrics is used as a form of identity access management and access control.

In a multiuser system or network, without password, anyone could log as a legitimate user and access to his/her confidential information. A system needs to provide different mechanisms for identity and authentication. For more precisions, many solutions for identification and authentication are detailed in chapter 10 of [Menezes 97].

After a successful authentication, the system constrains user's access only to the allowed resources. In general, a token or ticket is used for this purpose. Thus, the user ability to roam freely throughout the system is limited. This procedure is called authorization.

1.2 Problem Statement and Contribution

The identity is used in different types of networks as in Home Network, Cellular Network, etc... but this identity is used in the majority of the cases as a login to access to the system and not as a principal parameter in the cryptographic protocol i.e; the identity is not used by the system to generate the authentication keys.

The utilization of the identity depends on the type of networks, their managers and their security policies. We describe in the list below the major security risks in three different networks:

- In *Home Network Scenario*, the user at his/her home has to: protect the confidentiality of his/her files, protect the system against intruders via the Internet, prevent hard disk failures, etc... The most important issue is the protection of user privacy.

– In *Cellular Operators Scenario*, the subscribers needs to be authenticated and authorized to access to the services. The authentication of the users avoids the repudiation of involvement in a communication. For the Cellular Operator, the protection of their financial benefits and the privacy of their subscribers are the most important security goals.
– In *Government Network Scenario*, there are many agencies specialized in different security fields like police, army, border controls, secret services, etc... Each one of them has different security requirements in order to preserve the security of the citizens. We consider in this dissertation the border control scenario. Since many travelers enter and leave the country each day, the governments need to protect themselves from ID hackers and black listed criminals. The use of the electronic passport (e-Passport) aims to enhance the tracking of these outlaws. The major security problems related to the use of e-Passports are the protection of traveler's privacy, identity theft and identity fraud.

In these three cases, the identity is exploited in the authentication mechanisms only to identify the users. In our contributions, we aim to enhance and personalize those mechanisms by generating cryptographic keys depending on the user identity. Hence, we consider in this work two types of identity: the biometrics and any public string as email addresses.

In the first place, we combined biometrics with cryptography since biometrics are not sufficient alone to authenticate the user. We used biometrics to personalize and enhance the authentication mechanisms in personal Home Network scenario and in e-Passport authentication protocol. In the e-passport case, we used fingerprint or iris code to authenticate the traveler. In the latter contribution, we generated the cryptosystem parameters using the biometrical data.

In the second place, we used email addresses of the subscriber in the service authentication phase of the IP Multimedia Subsystem (IMS) to generate public/private key pair. This is done by choosing Identity Based Cryptography (IBC) in spite of the Authentication and Key Agreement (AKA) protocol. In the end, we focused on the performance of the signature verification and how to reduce it using the Batch Verification Scheme.

1.3 Organization of this Thesis

The aim of this dissertation is not to produce finalized protocol suites but to investigate how authentication protocols that use different available types of identity could be designed. This dissertation is composed by seven chapters. For the sake of readability, there are a two states of the art, one about cryptography(chapter 2) and the other one about the biometrics (chapter 3). Chapter 4 describes a solution to realize personalized access in the Home Network (HN). And chapter 5 concerns the use of biometrics in the third generation e-Passport. We have designed a new version of protocol using either fingerprint or iris code. Chapter 6 contains a rapid state of the art concerning the IMS and its authentication methods. The solution presented here uses Identity Based Cryptography (IBC) in the service authentication in the IMS. The solution is called IMS-IBC. In the end, we conclude this dissertation and we give some perspectives of future work in chapter 7.

Chapter 2

Cryptography

2.1 Introduction

Cryptography is specified as the science of information security. Modern cryptography has the following main four objectives:

1. *Confidentiality*: the information is protected against unauthorized disclosure
2. *Integrity*: the information, that was sent, has not been changed when it was stored or exchanged between sender and legitimate receiver
3. *Non-repudiation*: one of the entities involved in a communication cannot deny having participated in all or part of the communication
4. *Authentication*: the sender and the receiver can be convinced about each other's identity and the origin/destination of the information

To fulfill some or all of the above criteria, we need procedures and protocols that are known as cryptosystems. Cryptosystems can refer to mathematical procedures and computer programs. But, they also involve the measures taken by the cryptographer, like:

- preferring hard-to-guess passwords
- turning off systems that are no longer used
- not talking about sensitive procedures with strangers or outsiders to the system

In the following sections of this chapter, we introduce some basic concepts of security and cryptography. We choose Alice as the sender of the message and Bob as its receiver.

2.2 Algebraic Notations

A *group* is a set G together with an operation "*" that combines any two elements a and b to form another element of G denoted $a * b$. G contains an identity element e such $e * a = a * e = a$ and for each $a \in G$, there exists an element $b \in G$ such that $a * b = b * a = e$.

A *cyclic group* G is a group that can be generated by a single element a (the *group generator*). $\mathbb{Z}_n$ is a cyclic group of finite group order n and its generator a satisfies $a^n =$ e. If b is an element of $\mathbb{Z}_n$, then b can be written a^k, $\forall$ 0 <= k <= n-1.

A *field* F is a commutative group as for two compatible operations, addition and multiplication.

A *finite field* has a finite number of elements. The number of these elements is called the *order* of the field. As an example, the finite field $\mathbb{Z}/p\mathbb{Z}$ has p elements, usually labeled 0, 1, 2, ..., p-1, where operations are done modulo p.

2.3 Symmetric Cryptography

Symmetric cryptography uses the same private key to encrypt and decrypt data. Everyone who has the private key, can use the cryptosystem. Generally, symmetric key ciphers are known as block ciphers.

Symmetric cryptography algorithms are fast and adequate to process large streams of data.

The disadvantage of symmetric cryptography is the assumption to have two parties which have agreed on a key and to exchange that key in a secure manner prior to communication. Therefore, symmetric algorithms are often mixed with public-key algorithms in order to have secure and faster cryptosystem.

In figure 2.1, Alice and Bob must share an identical secret key for encryption and decryption. Alice encrypts a message M. The ciphertext C is sent to Bob. The latter decrypts C using the shared secret key to retrieve M. The fact to share the same key preserves the confidentiality of messages.

Some of the well known symmetric protocols are *Data Encryption Standard* (DES), *Advanced Encryption Standard* (AES) [stallings 03] to achieve confidentiality. Alice and Bob may also use a *Message Authentication Code* (MAC) algorithm such as Hash-based MAC (HMAC) [Bellare 96] to achieve data integrity and data origin authentication.

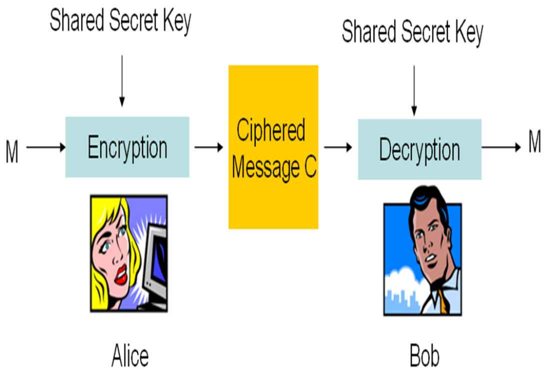


Figure 2.1: Symmetric System

2.4 Asymmetric or Public-key Cryptography

Public-key cryptography is also called asymmetric cryptography. It uses a couple of key, one private that must be hidden from unauthorized users and a public key available to anyone. The private key depends on the public key through a mathematical equation. The ciphered data using the public key can be deciphered only by the private key, and the signed data using the private key can only be checked with the public key.

The public key can be published to anyone. In a communication session, Both keys are unique.

The Public-key cryptography has two main branches:

- Public key encryption: Alice sends an encrypted message with Bob's public key. Upon receipt, Bob decrypts it with its private key. Figure 2.2 describes a scenario of encryption.

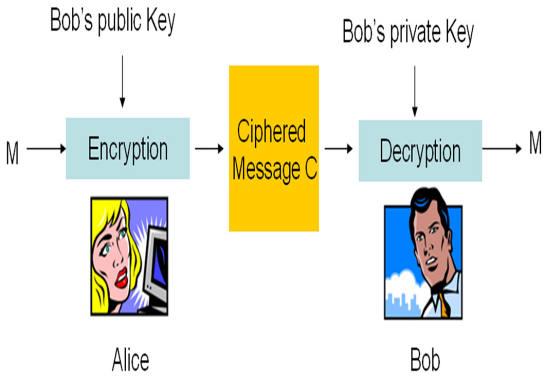


Figure 2.2: Asymmetric System

– Authentication: Alice sends a message encrypted with her private key, Bob decrypts it with Alice's public key. This scenario is well know as Digital signature that we detail in Section 2.5. Figure 2.3 presents a scenario of authentication.

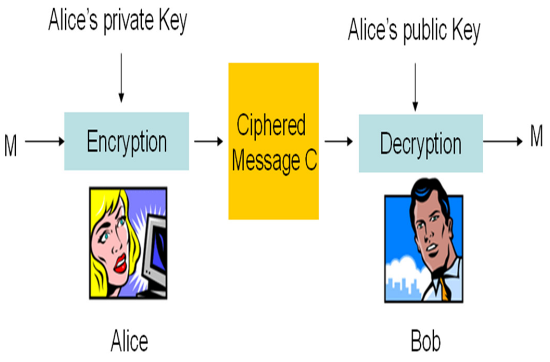


Figure 2.3: Authentication

The Asymmetric Cryptography is used also with digital certificates which contain the public key of the certificate owner. The private key is stored in a safe place by the owner. An application of such certificates is the implementation of a Public Key Infrastructure (PKI) to manage the authentication and digital signature of the owner.

The Asymmetric cryptographic protocols are based on two problems:

2.4.1 The Discrete Logarithm Problem (DLP)

The *Discrete Logarithm Problem* (DLP) is used in various public key infrastructure algorithms, such as Diffie-Hellman and ElGamal [stallings 03]. This problem has been studied for many years and cryptography based on it was strong against many forms of attacks.

The following applies to finite fields. We suppose that we have a prime number P (a number that is not divisible except by 1 and itself, P). This P is a large prime number with length not less than 300 digits. We have also two other integers, a and b used to compute N as follow:

$$N = a^b \bmod P, \text{ where } 0 <= N <= (P - 1)$$

This is equation is called *discrete exponentiation* and it is easy to compute. However, if we are given P, a, and N to find b, then we face a very hard problem.

2.4.2 The Integer Factorization Problem (IFP)

The *Integer Factorization Problem* (IFP) is one of the most fundamental of all mathematical concepts. We have two large prime numbers, P and Q. We multiply P and Q to get N. The problem is described as, being given N, how to retrieve the original P and Q? The *Rivest-Shamir-Adleman* RSA [stallings 03] encryption protocol is based on this problem. In such system, the public key is N and the the private is the couple P and Q numbers. We remember at the end, that the IFP has been studied intensely for the past 20 years and no solution that can factor it in a polynomial time is found yet.

2.5 Digital Signature

Sometimes, we send a message with a signature to verify the integrity of the message. The idea is to send a message digest obtained when hashing a message using a hash function to the message (eg, Secure Hash Algorithm (SHA)). Hash algorithms are one-way mathematical algorithms that have an arbitrary length input and produce a fixed length output string. A hash value is a unique and a compressed numerical representation of a piece of data. For example, MD5 produces 128-bit hash value for instance. It is unlikely to have collusion and find two distinct inputs that have the same hash value. The process steps are shown in Figure 2.4.

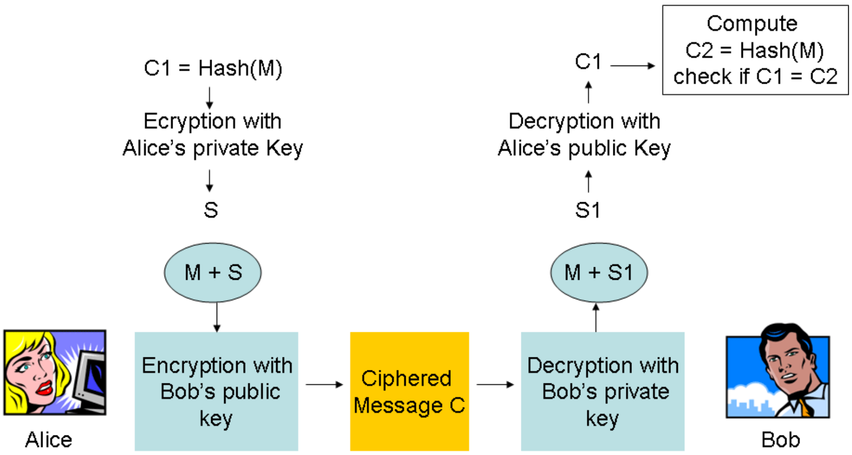

C1 = Hash(M)
Ecryption with Alice's private Key
S
M + S
Encryption with Bob's public key
Ciphered Message C
Decryption with Bob's private key
M + S1
S1
Decryption with Alice's public Key
C1
Compute C2 = Hash(M) check if C1 = C2
Alice
Bob

Figure 2.4: Digital Signature

This digest *C1* is encrypted with the Alice's private key to create a signature S. The message M and the signature S are encrypted by Bob's public key. Upon receipt, Bob decrypts the received message using its private key. He can extract the digest *C1* after decrypting the signature S using Alice's public key. He calculates a new digest *C2* from the received message M and he compares the two digests. If they are equal, then the message M was not modified.

In 2005, a widely-used cryptography algorithm, known as SHA-1, has been broken by three researchers at Shandong University in China [Wang 05b], [Wang 05a]. The actual attack called " collusion attack" reduced the complexity of breaking the SHA-1 standard to 2^{69} from 2^{80}. The complexity is a measure of the number of calculations that have to be performed to find a collision defined by two documents or files that produce the same hash. The attack is still at theoretical phase. In practice, the attack would be performed thousands of years on a most performant personal machine, and would still be slow even if we use a Grid Network.

2.6 Elliptic Curve Cryptography (ECC)

Cryptography mechanisms based on elliptic curves depend on arithmetic using the points of the elliptic curve. There exit two types of elliptic curves, one defined over a finite field $\mathbb{F}_p$, where p is a prime number and the other type is defined of $\mathbb{F}_{2^m}$ which is generated with the irreducible polynomial of degree m.

2.6.1 Elliptic Curve over $\mathbb{F}_p$

We assume first that $\mathbb{F}_p$ has characteristic greater than 3, where p is a prime number. An elliptic curve E over a finite field $\mathbb{F}_p$, is the set of all points (x, y) $\in \mathbb{F}_p * \mathbb{F}_p$ that verify the equation

$$y^2 = x^3 + Ax + B, \tag{2.1}$$

where A, $B \in \mathbb{F}_p$ satisfy $4A^3 + 27B^2 \neq 0 \bmod p$. There is another point O in this set called the point at infinity. This curve is denoted by E($\mathbb{F}_p$) or shortly by E. A point P of prime order in E($\mathbb{F}_p$) has the form $P(x_P, y_P)$ where x_P and y_P in $\mathbb{F}_p$ [Hankerson 04].

In [Menezes 93], it is reported: “The well-known theorem of Hasse states that

$$\text{card}(\text{E}(\mathbb{F}_p)) = p + 1 - t, \text{ where } |t| <= 2\sqrt{p}.$$

If the characteristic of $\mathbb{F}_p$ is 2 or 3, then a curve over $\mathbb{F}_p$ is supersingular if and only if it has j-invariant equal to 0.

An example of elliptic curve is presented in Figure 2.5.

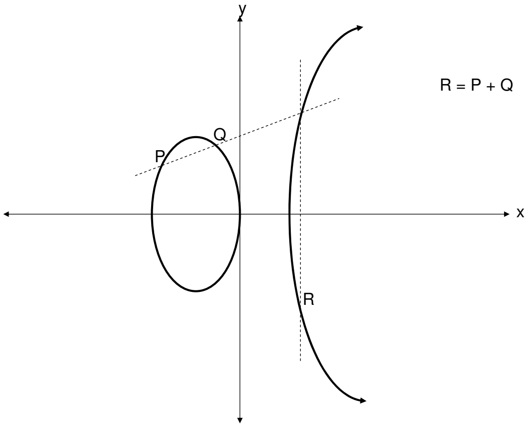


Figure 2.5: Elliptic Curve

The sum of two points in E($\mathbb{F}_p$) is also a point in E($\mathbb{F}_p$). Given two points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ on the curve E:

– If P = (x_P,y_P) ∈ E, then -P = (x_P, $-y_P$)

– If Q = -P, then P + Q = O

– If $x_P \neq x_Q$, then $R = P + Q = (x_R, y_R)$, where

$$x_R = s^2 - x_P - x_Q,$$
$$y_R = s(x_P - x_R) - y_P,$$

and

$$s = \frac{y_Q - y_P}{x_Q - x_P}$$

– If P = Q and $y_P = y_Q \neq 0$, then $R = P + P = 2P = (x_R, y_R)$ where

$$x_R = s^2 - 2x_P,$$
$$y_R = s(x_P - x_R) - y_P,$$

and

$$s = \frac{3x_P^2 - A}{2y_P}$$

Hence, the multiplication of a point P by a digit x gives a point $Q = x.P$ belonging to E. The multiplication by x times is translated into x additions i.e; $Q = \underbrace{P + P + ... + P}_{xtimes}$. To obtain $x.P$, the multiplier x is binary bit represented and then the techniques of *double and add* is used as follows: We choose as example x=$(9)_{10}$=$(1001)_2$=$(b_3b_2b_1b_0)_2$ where b_0=1, b_1=0, b_2=0, and b_3=1

This means that: 9 P = 2^3 P + 2^0 P = 8 P + P

Thus we obtain 2 P = P + P, then we obtain 4 P = 2 P + 2 P, and finally we obtain 8 P = 4 P + 4 P. To obtain 9 P, we add P + 8 P.

2.6.2 Elliptic Curve over $\mathbb{F}_{2^m}$

In this case, the equation of the elliptic curve is defined on a binary field $\mathbb{F}_{2^m}$ by

$$y^2 + xy = x^3 + Ax^2 + B, \tag{2.2}$$

where $B \neq 0$. The elements of the finite field are at most m-bit integers. These numbers represent a binary polynomial of degree $m - 1$ since the coefficients of the binary polynomial can only be 0 or 1. To have a secure cryptosystem, the parameter m is chosen to have a finite large number of points on the elliptic curve. Operations on binary polynomials are modulo irreducible polynomial of degree m. The latter generates the field $\mathbb{F}_{2^m}$.

2.6.3 Use in Cryptography

The elliptic curves have been well known mathematical concepts for centuries, but their application in cryptography was only from few decades. The Elliptic Curves Cryptography (ECC) was first suggested by V. Miller [Miller 85] and N. Koblitz [Koblitz 87]. They believed that the *Discrete Logarithm Problem* (DLP) was harder for elliptic curves than for finite fields. The Elliptic Curve Discrete Logarithm Problem (ECDLP) is described as follows: "P is a point on the curve, if $Q = x.P$, where x is a digit, try to find x". It is hard for an attacker to retrieve x. The last record for ECDLP was set in 2009, a 112-bit prime ECDLP was solved by the *Laboratory for Cryptologic Algorithms* at the Ecole Polytechnique Federale de Lausanne (EPFL) (http://lacal.epfl.ch/page81774.html). They run their experiments for almost 6 month on a cluster of more than 200 PlayStation 3 (PS3) game consoles.

ECC requires the computation of the cardinality of the curves. Schoof [Schoof 95] gave fast solutions and described several practical improvements implemented by Atkin and Elkies. The Schoof Elkies Atkin algorithm (SEA) is the algorithm used in elliptic curve over a finite field.

The Elliptic curve key length is shorter than Rivest Shamir Adleman (RSA) key length. If the key for ECC is 256-bit long, then for RSA, it is 3072-bit long. In Table 2.1 (deplicated from [X9.62 99]), more key size values are given by the NIST guidelines. Furthermore, it was proven that ECC outperforms RSA Algorithm [Gura 04], [Qingxian 05].

ECC Key size (bits)	RSA Key size (bits)	Key size ratio	AES Key size (bits)
163	1024	1 : 6	
256	3072	1 : 12	128
384	7680	1 : 20	192
512	15360	1 : 30	256

Table 2.1: Equivalent key sizes for ECC, RSA and AES (NIST)

In this thesis, we chosen elliptic curves over $\mathbb{F}_p$. We describe in the following three ECC protocols.

2.6.3.1 Elliptic Curve Diffie-Hellman ECDH Key Agreement Protocol

This protocol is a new variant of the Diffie-Hellman protocol using Elliptic Curve Cryptography ECC. It is described in the Certicom Research report [Research 00]. This is how the algorithm is performed:

- Alice and Bob select an elliptic curve E defined over $\mathbb{F}_p$. The number of points in $\mathrm{E}(\mathbb{F}_p)$ should be divisible by a large prime n.
- They select a point $P \in \mathrm{E}(\mathbb{F}_p)$ of order n.
- Alice selects a statistically unique and unpredictable integer a in the interval [1, n-1]. Bob chooses the integer b in [1, n-1].
- Alice computes point $C = a.P$ and sends it to Bob.
- Bob computes point $D = b.P$ and sends it to Alice.
- Alice and Bob can now computes a common point $K \in \mathrm{E}(\mathbb{F}_p)$:
  $K = a.D = a.(b.P) = (a.b).P = b.(a.P) = b.C$

Durlanik et al. [Durlanik 05] made some experimental tests to compare ECDH and DH. They reported that “Besides of key sizes it can be said that ECDH is faster than DH by means of execution times and memory usage statistics according to the comparisons”. For example, the time needed to generate the Elliptic Curve Domain parameters with 256-bit prime is much lower that the one needed for DH Domain parameters with 512 bits (0.0676 seconds for ECDH-256 and 0.5783 seconds for DH-512).

There is another authenticated protocol for key agreement based on the DiffieŰHellman scheme. It is called Elliptic Curve Menezes–Qu–Vanstone (ECMQV). More informations about this protocol can be read in [Hankerson 04].

2.6.3.2 Menezes-Vanstone Protocol

Menezes and Vanstone [Menezes 93] designed a public key cryptosystem based on Elliptic Curves. This protocol is the version of the ElGamal encryption [ElGamal 85] using elliptic curves. It needs some initialization steps like:

- Alice and Bob choose an elliptic curve E defined over $\mathbb{F}_p$ where p is prime and $p > 3$
- Alice chooses a point $P \in E$ and s a positif integer. She computes the point $Q = s.P$
- Alice publishes the curve E and the points P and Q as her publics parameters. The number s is her private key.
- the plaintext x = *(y,z)* $\in \mathbb{F}_p * \mathbb{F}_p$ is a couple of integer modulo p. The plaintext x is not necessarily a point of *E*.

To encrypt the message x, Bob

- generates a random number k (k is fresh for each session)
- computes:

  $u = k.P$

  (c,d) $= k.Q$ (c and d are the coordinates of the points $k.Q$ of E)

  $v \equiv c.y \bmod p$

  w $\equiv d.z \bmod p$
- sends the ciphertext *(u, v, w) to Alice.*

To decrypt the ciphertext (u, v, w), Alice computes:

$s.u = s.(k.P) = k.(s.P) = k.Q = (c, d)$

$c^{-1}.v = c^{-1}.c.y \equiv y \bmod p$

$d^{-1}.w = d^{-1}.d.z \equiv z \bmod p$

In [Rahouma 09], the author provides a new modified variant of Menezes and Vanstone elliptic curve cryptosystem. This new variant uses many different curves, thus, there are separate cryptosystem related to each curve.

2.6.3.3 Elliptic Curve Digital Signature Algorithm (ECDSA)

The Elliptic Curve Digital Signature Algorithm (ECDSA) was developed by the the American National Standards Institute (ANSI) by the Accredited Standards Committee on Financial Services, X9 [X9.62 99]. Moreover, the ECDSA uses the Elliptic Curve Discrete Logarithm Problem (ECDLP). The latter is analog to the Discrete Logarithm Problem described in Section 2.4. The ECDLP is described as: Given $P \in \mathrm{E}(\mathbb{F}_p)$ and $Q = aP$, find a (1<= a <= n).

An elliptic curve E defined over $\mathbb{F}_p$ with large group $\mathrm{E}(\mathbb{F}_p)$ of order n and a point P of larger order are chosen by Alice and made public to all users. There are three different primitives:

- *ECDSA Key Generation* - Alice follows these steps:
  1. First, she chooses a random integer $d \in [2; n-2]$.
  2. Then , she calculates $Q = d.P$.
  3. In the end, she publishes her public parameters *(E; P; n; Q)* and she keeps safe her private key *d*.
- *ECDSA Signature Generation* - Alice signs a message m following these steps:
  1. She selects a random integer $k \in [2; n-2]$.
  2. She calculates $k.P = (x_1; y_1)$ and $r = x_1 \bmod n$. If $r = 0$ then she return to select a new k.
  3. She computes $k^{-1} \bmod n$.
  4. She computes $s = k^{-1}.(H(m) + d.r) \bmod n$. H is the secure hash algorithm (SHA-1). If $s = 0$, Alice needs to start from the begining.
  5. the pair of integers *(r; s)* are the signature for the message *m*.
- *ECDSA Signature Verification* - Bob verifies Alice's signature *(r; s)* on the message m by performing the following steps:
  1. He calculates $c = s^{-1} \bmod n$ and *H(m)*.
  2. He computes $u_1 = H(m).c \bmod n$ and $u2 = r.c \bmod n$.
  3. he computes $u_1.P + u_2.Q = (x_0; y_0)$ and $v = x_0 \bmod n$.
  4. Bob approves the signature if $v = r$.

Johnson et al. [Johnson 98] provide a comparison between ECDSA and DSA. They concluded that ECDSA has significant advantages over DSA like:

- The ECDSA prevents the weakness found in DSA which allows the selective forgery of one message if the adversary can choose the system parameters.
- In ECDSA, it is mandatory to check, during the signature genration, if the digital signature (r,s) is non-zero. Or, in DSA, it is optional.
- It is harder to recover a private key from a public key in case of Elliptic Curve Discrete Logarithm Problem (ECDLP) than using Discrete Logarithm Problem (DLP). As an example, for 163-bit prime number p (ECDLP), the attacker needs 9.6 * 10^{11} Millions Instructions Per Seconds (MIPS) years and for 512-bit key (DLP), the attacker needs 3 * 10^4 MIPS years.

2.7 Identity Based Cryptography (IBC)

The Identity Based cryptography (IBC) has emerged as a long-term evolution or substitution to Public Key Infrastructure (PKI). It is a cryptosystem in which the public key is generated using the identity of the entity (user) and the private key is the public key multiplied by the secret key of the server. The latter is responsible of the user's private key distribution and is called the Private Key Generator (PKG).

The IBC concept is old and was first proposed by Shamir in 1984 [Shamir 84]. Shamir's original motivation for identity-based encryption was to simplify the certificate management in e-mail systems. Then, the first fully practical and secure identity-based public key encryption scheme was presented by D. Boneh and M. Franklin in [Boneh 01], using the fundamental operations of Elliptic Curve Cryptography (ECC) and the bilinear pairing. Since then, the development of Identity based cryptosystem intensifies rapidly. As an example, D. Boneh co-founded in 2002 *Voltage Security* [Boneh 02]. In the next paragraphs, we present the bilinair pairing, the Identity Based Encryption and two others protocols based on IBC.

2.7.1 Bilinear pairing

Let $(G_1; +)$ and $(G_2; .)$ be two cyclic groups of prime order q. G_1 is an additive group and G_2 a multiplicative group. The bilinear pairing is given as e: $G_1 * G_1 \rightarrow G_2$, which satisfies the following properties:

1. Bilinearity: For all $P;\ Q;\ R \in G_1$;

   $e(P + Q, R) = e(P, R).e(Q, R)$ and $e(P, Q + R) = e(P, Q).e(P, R)$;

2. Non-degeneracy: If P is a generator of G_1, then, $e(P, P)$ is a generator of G_2 (Cyclic Group).

3. Computability: It is easy to compute $e(P,\ Q)\ \forall P; Q \in G_1$.

A bilinear map satisfying the three properties above can be considered as an admissible bilinear map [Boneh 01].

2.7.2 Identity Based Encryption (IBE)

The Identity-Based Encryption (IBE) is an important primitive of IBC. It is a kind of public-key encryption in which the public key of a user is generated using the identity of the user (e.g. a user's email address). The following functional routines are illustrated in Figure 2.6.

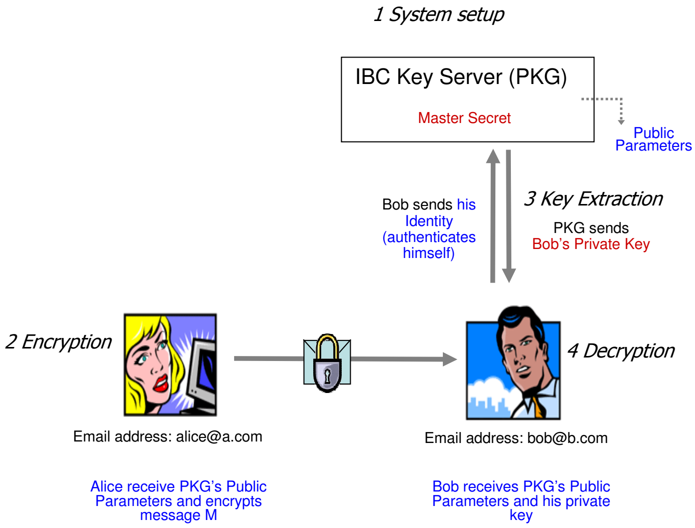

1 System setup
IBC Key Server (PKG)
Master Secret
Public Parameters
Bob sends his Identity (authenticates himself)
3 Key Extraction
PKG sends Bob's Private Key
2 Encryption
4 Decryption
Email address: alice@a.com
Email address: bob@b.com
Alice receive PKG's Public Parameters and encrypts message M
Bob receives PKG's Public Parameters and his private key

Figure 2.6: IBE's phases

1. **System Setup:** IBE systems rely on the Private Key Generator (PKG) which is a trusted central authority managing the system's parameters. The PKG generates its parameters, including a master secret S involving in user's private keys generation. The system parameters called "params" are: the prime number p, the order q, the generator point P, PKG's public point $P_{pub}= S.P$ and the MapToPoint which is the

hash function used by the PKG to convert a string into a point in the elliptic curve E).

2. **Encryption:** When a user (Alice) wishes to send an encrypted message to another user (Bob), she first generates Bob's public keyK_{pubBob}. Then, she encrypts a plaintext message M with K_{pubBob} to obtain cipher message C. In the end, she sends C to Bob.

3. **Key Extraction:** When Bob receives the message, he needs to decrypt it. He authenticates himself to the PKG in order to obtain the secret key $K_{privBob}$= S.K_{pubBob}. The latter is needed by Bob to decrypt the cipher message C.

4. **Decryption:** When Bob receives $K_{privBob}$, he decrypts the cipher message C to obtain the plaintext message M.

2.7.3 Identity-Based Signature Scheme

The Identity-Based Signature Scheme is an another primitive of IBC. It is a signature scheme in which the private key is generated by the PKG using the user's identity. The following functional routines are illustrated in Figure 2.7.

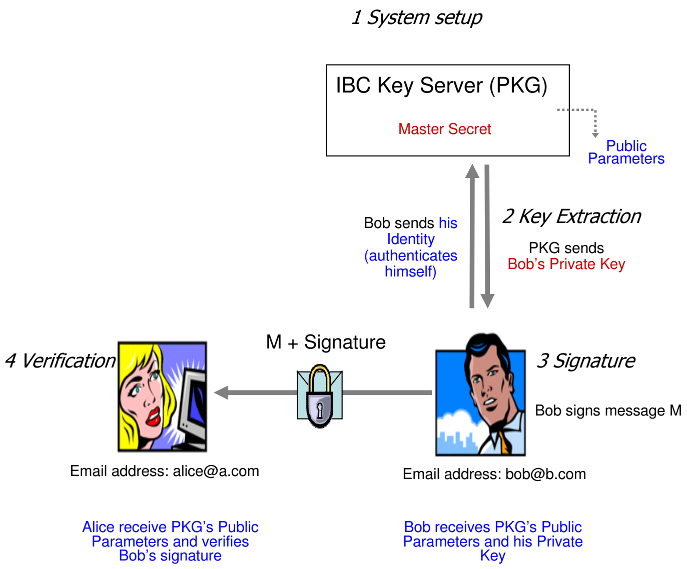

1 System setup
IBC Key Server (PKG)
Master Secret
Public Parameters
Bob sends his Identity (authenticates himself)
2 Key Extraction
PKG sends Bob's Private Key
M + Signature
4 Verification
3 Signature
Bob signs message M
Email address: alice@a.com
Email address: bob@b.com
Alice receive PKG's Public Parameters and verifies Bob's signature
Bob receives PKG's Public Parameters and his Private Key

Figure 2.7: Identity-Based Signature Scheme

1. **System Setup:** same as described in sec:IdentityBasedEncryptionIBE.

2. **Key Extraction:** Bob authenticates himself to the PKG in order to obtain the secret key $K_{privBob}$= S.K_{pubBob}. The latter is needed by Bob to decrypt the cipher message C.
3. **Signature:** Bob signs a message M using his private key $K_{privBob}$ (he hashes M and encrypt the hash value H_1 with its private key). Then, he sends the message M and its signature *Sig* to Alice.
4. **Verification:** When Alice receives the message M and the signature *Sig*, she hashes the message M and gets H_2. She then decrypts the signature *Sig* using Bob's public key that she computed using Bob's Identity and the PKG's public parameters. She gets the hash value H_1. Finally, she compare the two hashes and if they rae equal, then the signature is verified.

2.7.4 Certificateless Public Key Cryptography CL-PKC

CL-PKC was presented by Al-Riyami and Paterson in [Al-Riyami 03]. In CL-PKC cryptosystem, an entity called Key Generation Center (KGC) is responsible for generating a Partial Key to the user which has one and unique identity in the system.

The certificatless encryption runs five steps algorithm which are defined as follows:

- **Setup:** In this step, the Key Generation Center (KGC) generates its private *master-key*, the public parameters *params* and its public key using a security parameter k.
- **Partial Secret Key Extract:** The KGC generates Alice's partial secret $D_A \in G_1^*$ using its *master-key*, *params* and Alice's identity $ID_A \in \{1,0\}^*$ as inputs.
- **User key generation:** Alice generates first of all a secret value X_A, using *params* and her identity ID_A. Then, she takes her partial secret D_A, the *params* and her secret value X_A as input to generate the private key $S_A \in G_1^*$. In the end, she takes the *params* and her secret value X_A as input to generate the public key $P_A \in G_1^*$.
- **Encrypt:** Bob wants to send the message M ciphered with Alice's public key P_A. He ciphers the plaintext M using the *params*, Alice's ID_A and her public key P_A into the ciphertext C or $\perp$ (meaning encryption failure).
- **Decrypt:** After receiving the ciphertext C, Alice uses the *params* and her private key S_A to retrieve the plaintext M or $\perp$ (meaning decryption failure).

Although, CL-PKC has interesting advantages like the certificate implicitly, the Denial of Decryption (DoD) attack is still a relevant problem [Ahmad 09]. An attacker can change the legitimate user's public key by a fake public key. Although the attacker can not decrypt the received message, he/she does not let the legitimate receiver decrypt the message.

2.7.5 TIBC: Trade-off between Identity-Based and Certificateless Cryptography

In [Ahmad 09], the authors merged Identity Based Cryptography (IBC) and Certificateless Public Key Cryptography (CL-PKC) systems to solve the problem of user impersonation by providing an implicit certificate depending on two separate processes. They called their system Trade-off between Identity-Based and Certificateless Cryptography (TIBC). Three entities are involved in TIBC: a Key Generation Center (KGC), a publisher and his/her own Public Key generator (PKG). Based on the bilinear pairing, the TIBC is performed at two levels of interaction: (1) between KGC and the publisher, and (2) between PKG and the publisher.

– *Level-1:* After choosing an admissible bilinear map, the KGC generates *params* and it fellows these steps:

  1. **KGC_MasterKeyGeneration:** The KGC generates its private *master-key* and public key using the *params* and 1^y where $y \in \mathbb{N}$ is a security parameter.
  2. **PartialKeyGeneration:** The KGC generates Alice's partial key $PartialKey_A$ using its master key and Alice's identity $ID_A \in \{1,0\}^*$.
  3. **UserKeyGeneration:** Alice generates her private key $S_A \in G_1^*$ and her public key $P_A \in G_1^*$ using a secret X_A, *params*, $PartialKey_A$ and her identity ID_A.

– *Level-2:* In this level, Alice chooses a secret prime value n, and calculates the point $F = n.S_A \in G_1^*$ (hard Discrete Logarithm Problem) and sends F= (X_F, Y_F) to her PKG. Then, four steps are performed:

  1. **PKGSetUp:** After receiving the point F, the PKG chooses an admissible bilinear map, a point $P' \in G_1^*$, the abscissa X_F of the point F as its master secret key and then generates a Master Public Key $P''_{pub} = X_F.P'$.
  2. **ExtractUserPublicKey:** the PKG computes the point $Q_{ID_A} = H_1(ID_A)$ which is Alice's final public key. H_1 is a MapToPoint Function.
  3. **ExtractUserPrivateKey:** The PKG computes Alice's final private key $U_A = X_F.Q_{ID_A}$
  4. **Encryption/Decrytion:** These operations are the same defined for the IBC.

The most important advantages of TIBC is the "self-generated" certificate since the user's private key depends on a secret value given by the user himself.

2.8 Conclusion

In this chapter, a comprehensive review of cryptographic protocols is presented. We remind that we need to be conform to the Cryptographic Key length Recommendation [BlueKrypt 10] Since the cryptanalist are working hard to improve their capability for attacking systems. We started this state of the art by describing the symmetric and asymmetric systems. The basic cryptographic procedures are: the encryption, the decryption and the signature. The Discrete Logarithm Problem (DLP) make these systems efficient against prime factorization. Then, we introduced the Elliptic Curve Cryptography (ECC) which aims to enhance the performance of cryptographic systems and make them hardly breakable. This is due to the Elliptic Curve Discrete Logarithm Problem (ECDLP). We described only three cryptographic protocols based on ECC which are: Elliptic Curve Diffie-Hellman (ECDH) Key Agreement Protocol, Menezes-Vanstone Protocol and Elliptic Curve Digital Signature Algorithm (ECDSA). After that, we introduced the Identity Based Cryptography (IBC) which is a type of asymmetric cryptographic system. The IBC uses a publicly known string belonging to an individual or organization as a public key. The public string could be an email address, a domain name, or a physical IP address. We presented different variant of solution using IBC, like Identity Based Encryption (IBE), Identity-Based Signature Scheme, Certificateless Public Key Cryptography (CL-PKC) and TIBC: Trade-off between Identity-Based and Certificateless Cryptography. The IBC is primordial in our work since it is used in the proposed solutions.

Chapter 3

State of the Art of Biometrics

3.1 Introduction

The biometry or biometrics aims to improve human ability to identify a person. Different biometrical techniques are at present under research, including fingerprints, facial, palm prints, retinal and iris scans, hand geometry, signature capture and vocal characteristics. However, its use raises issues that affect the people's privacy. One innovative idea is to combine biometry with cryptography in order to generate more secure secret key dependent of the human body.

In this chapter, we present a state of the art in biometrical technologies and biometric cryptosystems.

3.2 Biometrics

Biometrics identify a person using its own identifiable and verifiable data. It answers the question: "what one is?" (fingerprints, hand, face ...). In Table 3.1(replicated from [Itakura 05], page 291), the authors present an overview of the most used biometrical technologies.

Biometrical authentication is the verification of human identity using measurement of biological characteristics. Biometrical authentication has become internationally recognized as a mean for people to authenticate themselves to computing systems. There are two performance thresholds of biometrical authentication mechanisms:

- False Acceptance Rate (FAR) is the percentage of false acceptances among the identification attempts.

– False Rejection Rate (FRR) is the percentage of false rejections among the identification attempts.

We shall add another parameter: the Equal Error Rate (EER) which is the point where the probability of false acceptance and that of false rejection become equal. In the literature, The Genuine Acceptance Rate (GAR) is used instead of FRR. we just have to know that GAR = 1 - FRR.

Biological information	**Fingerprint**	**Iris**	**Face**	**Voice print**	**Signature**	**DNA**
Identifying principle	minutiae	iris patterns	facial features	vocal sounds	Difference in handwritten letters pressure, timing	short tandem repeats
FAR	2.10^{-6} or less	$8,3.10^{-7}$ or less	10^{-2} or less	3.10^{-2} or less	10^{-2} or less	10^{-15} or less
FRR	0.05% or less	0.1% or less	1% or less	3% or less	1% or less	Less than measuring error
Sensor	Image sensor	Camera	Camera	Microphone	Pressure sensor	DNA analyzer
Data size of templates in bytes	250 to 500	250	1000	1000	1000	20
features	Small size, economic and high precision	Small psychological stress and high precision	Small psychological stress	Small psychological stress	High precision in dynamic signature	High precision, uniqueness, and high stability with time
Problem	Degradation of fingerprint due to dried skin	–	Change due to aging,camera angle, hat, or eye glass	Voice Change in puberty or due to thirsty throat	Ease of Imitation	Long analyzing time, high price, and privacy concerns
Risk of unauthorized use	Fingerprint marked	Eye captured by camcorder	Face captured by camcorder	Voice recorded by microphone	handwriting imitated	stolen hair with root

Table 3.1: Summary about Biometrical Algorithm Characteristics

In [Cavoukian 07], authors gave some performance values related to FAR and FRR: “For most biometric systems, FRR ranges from 0.1% to 20%, meaning that a legitimate user is rejected from one out of 1000 times to one out of five times on average. FAR ranges from one in 100 (low security applications) to one in 10,000,000 (very high security applications)”.

Research works are still underway to stabilize the biometrical data. To perform authentication, it is better to couple it with a smart card or a secure token (small storage element with high resistance to attacks, even physical). In the enrollment phase, the system stores the biometric template in a smart or Radio Frequency Identification (RFID) card. In the verification phase, the user provides a Personal Identification Number (PIN) code to unlock the access to the biometric data. He/She enrolls his/her fingerprint which is compared to the one stored in the card. This system is well used in sensitive nuclear stations or in airports. We describe in the next sections others methods to deploy biometric data in cryptographic system.

The biometrical authentication systems consist of two steps, which are the enrollment and the authentication. In the *enrollment phase*, the biometrical data are captured from a user, where the biometrical features are extracted and eventually stored in a database. As for the *authentication phase*, we need to distinguish two alternative methods:

– *Verification*: the user, wishing to be recognized by the system as the holder of an identifier, presents some form of identifier (like used ID, Automatic Teller Machine (ATM) card) and a biometrical characteristic. The new biometrical feature is compared with the stored ones, associated with the provided identifier (1-to-1 matching);
– *Identification*: In the case where the user does not provide an identifier, the system needs to find the user's identifier. The extracted biometrical feature is compared to the entire database for matches (1-to-N matching).

The difference between a password and a biometric template is relative to the replay attack, a password is supposed to be secret, while biometric templates are not. Some systems incorrectly assume that biometric measurements are secret and grant access when matching biometrical features are presented. To solve such a problem, one of the best way is to biometry with cryptography.

A classical biometric system is shown in Figure 3.1 presenting the enrollment and the verification phases. In such systems, pertinent biometrical features are extracted from the user's biometrical data (e.g., minutia from the fingerprint, iris code, etc...). These features need to be stored in form of templates in a central database or on personal possessions (such as e-Passport) for future comparison. The user verification is carried out by comparing the stored template with the newly provided biometrical data by the user.

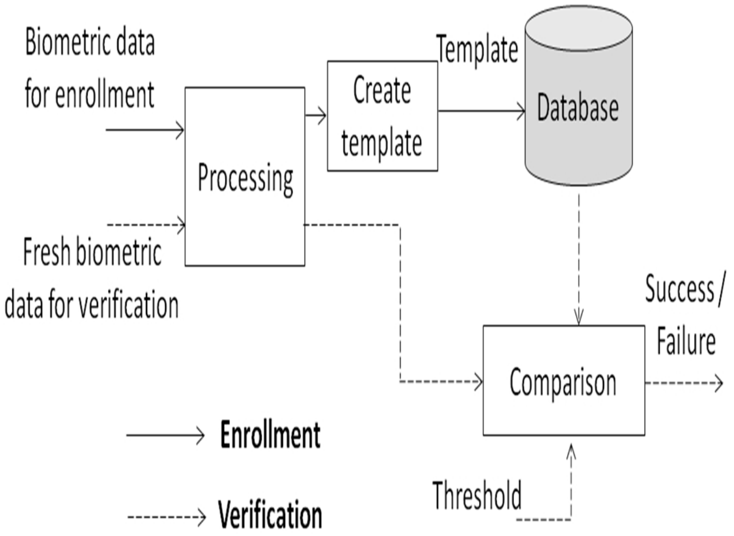


Figure 3.1: Classical Biometrical System

There are some problems associated with such classical biometric verification systems:

- The biometrical characteristic is permanently associated with the user. Hence, it is not possible to issue different templates for different applications for the same user. The biometric data stolen from one biometric system can be used to attack another system based on the same biometrics.
- Moreover, even if the compromise is detected, that data cannot be replaced and becomes unusable by the system. This is called non-revocability of biometrics.
- Another problem is that the classical biometric comparison results only in one-bit information - success or failure. Such system can be attacked by a Trojan horse which can replace the biometric verification module and induce the desired result bit in the system.
- The storage of biometrical data in databases by non governmental authorities meets opposition from the French Data Protection Authority *Commission Nationale de l'Informatique et des Libertés* (CNIL) [CNIL 07] and others organizations worldwide.

In the following, we present three solutions to enhance biometrics results. The first one is the use of multimodality in biometrics, the second one is the cancelable biometric and the third one is the combination of cryptography and biometry.

3.3 Multimodal Biometrics

Systems that use a single biometric face a variety of problems such as: fuzzy data, the intra-class variations, degrees of freedom restricted, non-universality, identity theft, and an unacceptable error rate. Some of these limitations can be avoided by choosing multimodal biometric systems. Theses latter are based on using multiple biometrics at the same time. The multimodal biometrics requires a merging mechanism between different modalities.

Arun Ross and Anil K. Jain [Ross 04], provided a general statement on already existing multi-biometric solutions. They described the system's deficiencies using a single modality. Then, they presented three merging levels for the multimodality presented in Figure 3.2.

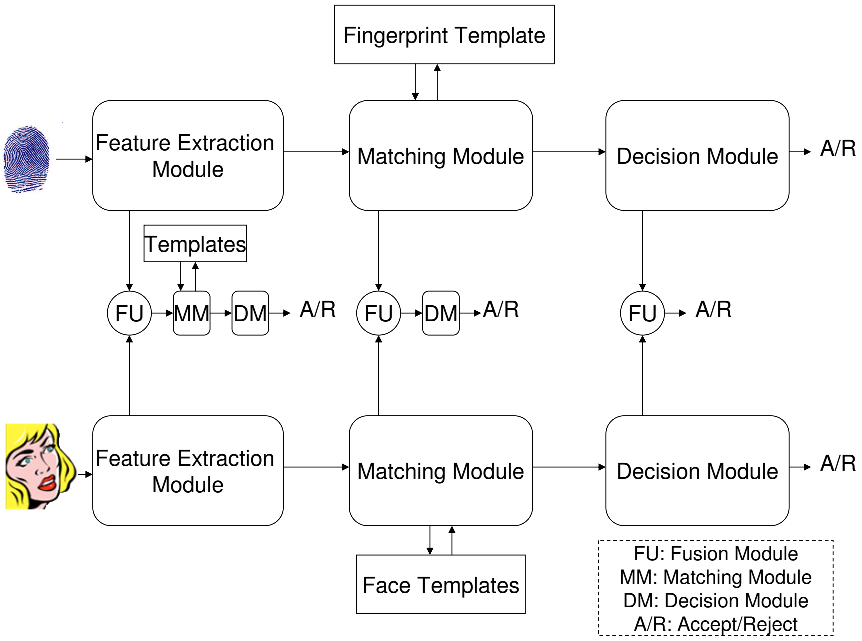


Figure 3.2: Multimodal System

There are three possible fusion levels:

– Fusion at the Data or Feature Level

– Fusion at the Match Score Level

– Fusion at the Decision Level

3.3.1 Fusion at the Data or Feature Level

The templates are put in the same form to be merged directly after the enrollment phase. At this fusion level, there are 2 problems. The first is that all biometric sensors may not be compatible (eg facial Eigenfaces and fingerprint minutiae). The second is that most commercial biometric systems do not allow access to their used process (or raw data).

3.3.2 Fusion at the Match Score Level

At this level, each module provides a comparison score. This is provided after a normalization and fusion score phases. This method is the most used since it is easy to calculate a final score for the decision.

In [Snelick 03] and [Snelick 05], standardization and fusion methods are presented. The output of biometric sensors is S and the normalized score is N. In Table 3.2 (from [Snelick 03] page 69) , the authors present the standard normalization methods.

$Min - Max(MM)$	$N = (S - min)/(max - min)$
$Z - score(ZS)$	$N = (S - moyenne/variance)$
$Tanh$	$N = 1/2[tanh(0.01(S - moyenne)/variance) + 1]$

Table 3.2: Summary of Normalization Techniques

In the following, N_i are the normalized scores and $P(genuine/N_i)$ is the probability that score is provided by a legitimate person and not by an impostor. The fusion techniques are showed in Table 3.3 (replicated from [Snelick 03] page 70).

Afterwards, the authors compare the final scores obtained from single biometrical modal and from the association of normalization methods and different fusions. The use of the latter techniques enhances the performance significantly over the single-modal face or fingerprint minutiae. For example, with a False Acceptance Rate (FAR) fixed at 0.1%, the simple sum fusion with the min-max normalization has a Genuine Acceptance Rate (GAR) equals 94.9%, which surpasses that of face, 75.3%, and fingerprint, 83.0%.

Simple Sum	$\sum_{i=1}^{N} N_i$
Minimum Score	$Min(N1, N2, ..., Nn)$
Maximum Score	$Max(N1, N2, ..., Nn)$
Sum of Probabilities	$\sum_{i=1}^{N} P(genuine/N_i)$
Product of Probabilities	$\prod_{i=1}^{N} P(genuine/N_i)$

Table 3.3: Summary of Fusion Techniques

3.3.3 Fusion at the Decision Level

At this level, there is a vote among the decisions to accept or reject the candidate. This method is difficult to realize since there are not enough data.

3.4 Cancelable Biometrics

To generate strong cryptographic keys from biometrical data, we need to find a mapping technique. Biometrical techniques face a major problem which is their complex reinitialization. When a credit card number is stolen, the bank assigns a new credit card number to its client. When biometric features are stolen, no substitution is possible because the new enrolled features are the same as the stolen ones.

The concept of “cancelable biometrics” was suggested to avoid this problem and, by the way, offers biometric template protection. Figure 3.3 shows both the enrollment and the identification phases when cancelable biometrics are used.

Cancelable biometrics rely on a repeatable distortion of the biometric features. This action protects the sensitive user’s biometrical data. If a cancelable feature is lost or misused by an attacker, it is easy to change the distortion characteristics. In this case, the same biometrical data are used to generate a new cancelable biometrics. In general, the transformation used for the distortion are non-invertible (i.e; if an attacker succeeds in stoling the cancelable biometrics, he/she cannot recover the real biometrical data).

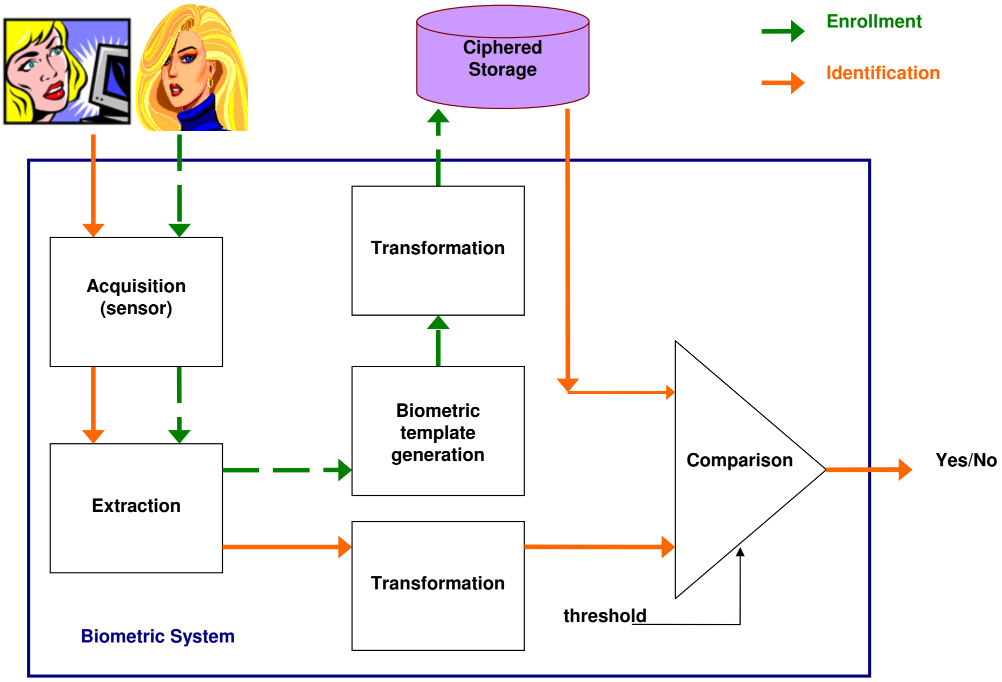


Figure 3.3: Cancelable Biometrics

The concrete idea of cancelable biometrics was suggested by Bolle et al. [Bolle 02]. The BioHashing method was firstly presented by Jin et al. [Jin 04]. The basic idea of BioHashing is based on the generation if an orth-normal matrix using a hash key. The biometrical feature vector is projected onto the matrix to create a BioHadh. Lumini et al. [Lumini 07] improved this method by adding some permutations and normalizations.

A realization of non-invertible transformation was reported by Ratha et al. [Ratha 07] where fingerprint data is transformed by a sequence of three non-invertible transformation functions which are based on *cartesian*, *polar* and *surface folding transformation* of the minutiae positions.

The best results were obtained using surface folding transformation where both the position and the orientation of the minutiae are modified. the Figure 3.4 (replicated from [Ratha 07] page 567) presents a cancelable fingerprint obtained using this method.

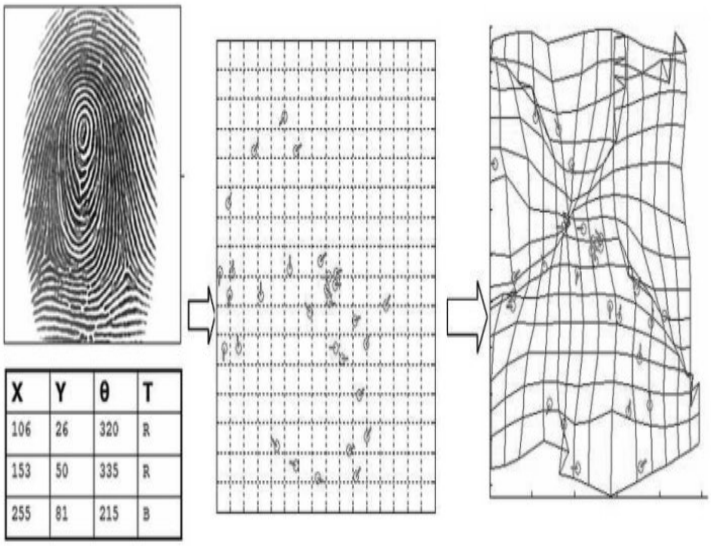


Figure 3.4: Cancelable Fingerprint

In Table 3.4, we present a summary of the solutions described above.

Techniques	Experiments	Results
bolle [Bolle 02]	theoretical	–
BioHashing [Jin 04]	Fingerprint, FVC 2002 (Set A)	ERR ≈ 0
BioHashing [Lumini 07]	Faces (ORL and Yale-B Databases) Signatures (SUBCORPUS-100) Fingerprint (Fingerprint, FVC 2002)	ERR $>= 2.4$ ERR ≈ 0 ERR $>= 0.4$
non-invertible transformation [Ratha 07]	188 fingerprint pairs (IBM-99 Database)	FRR: 5% / FAR: 0.1%

Table 3.4: Cancelable Solutions

In the next section, we present the biometric cryptosystems' state of the arts.

3.5 Biometric Cryptosystems

A biometric cryptosystem uses biometrical data to reinforce existing key or to participate in the cryptographic key's generation. The first and easiest solution was to encrypt the biometric template and to send it in a communication session. In this section, we do not address this kind of solution but the solutions involving directly the biometrical data with cryptography. We organize the solutions found in our readings following three criteria:

1. Locking the cryptographic key using a biometric template;
2. Using Error Correcting Codes;
3. Using biometric template as part of the key.

We define A as the biometric template on the enrollment phase and B as the template during the verification phase. In all these solutions, the key is independent of the biometric template which means that it is chosen from the beginning and it won't be changed with time.

3.5.1 Locking the Key using Biometric Template

This section describes some solutions aiming to reduce the fuzziness of biometrical features (the fuzziness is caused by some uncertainties and errors in the enrollment phase).

3.5.1.1 Soutar et al.'s Solution

Soutar and his colleagues [Soutar 99] proposed a Biometric Encryption algorithm which links a biometrical data (such as fingerprint) with a conventional cryptographic key. This algorithm can retrieve the key using a new enrolled biometrical data. This algorithm combines a cryptographic key (typically 128 bits) with the user fingerprint image. The key is successfully recovered only by using filters based authentication (using correlation functions). They assume that the multiple enrolled fingerprints are lined up in advance. Figure 3.5 shows the enrollment phase and Figure 3.6, the verification phase. These two figures where replicated from [Soutar 99] page 15 and 16.

The enrollment phase has three steps:

E-1 *Image Processing*: The processor merges a list of input fingerprint images with a random array to outputs two arrays: $H_{stored}(u)$ and $C_0(X)$.

E-2 *Key linking*: It links a cryptographic key, k_0, to the value, $c_0(x)$, using the link algorithm.

E-3 *Identification code creation*: It derives from the key, k_0 an identification code, id_0.

At the end, a secure data block called Bioscrypt is created, it contains; $H_{stored}(u)$, a look up table and id_0.

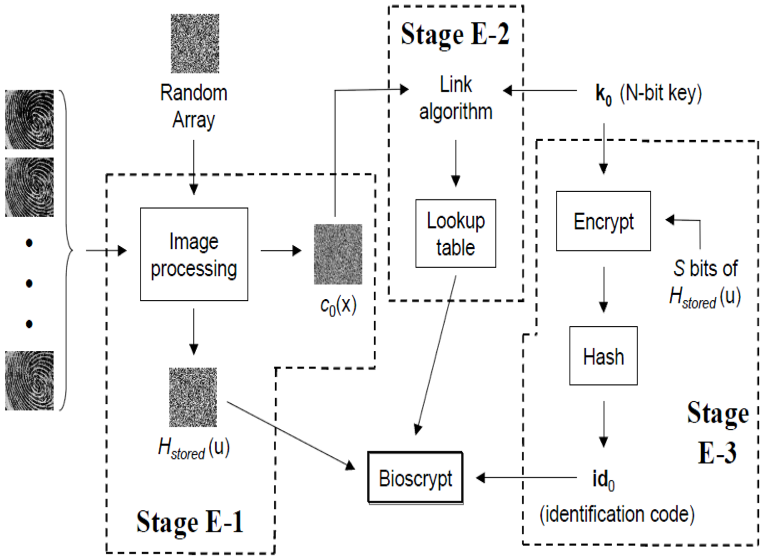


Figure 3.5: Enrollment Phase

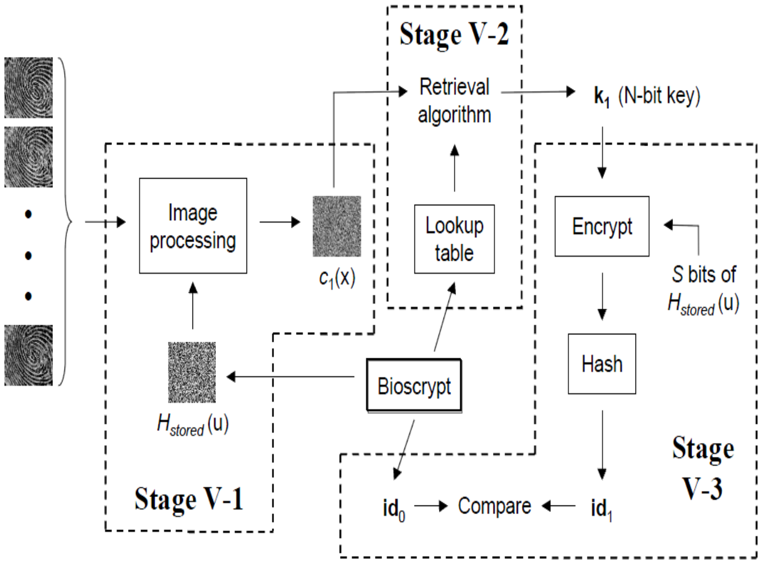


Figure 3.6: Verification Phase

V-1 *Image Processing*: The processor merges $H_{stored}(u)$, from the Bioscrypt, with a new list of input fingerprint images to create an output pattern, $C_1(X)$.

V-2 *Key Retrieval*: It gives $c_1(x)$ and the look up table as input to the retrieval algorithm to retrieve a key, k_1.

V-3 *Key Validation*: It generates a new identification code, id_1 and compares it with id_0 to validate k_1.

3.5.1.2 Fuzzy Vault

The early contribution using this method was done by Juels and Wattenberg [Juels 99]. In this work, they proposed a fuzzy commitment scheme. Then, Juels and Sudan [Juels 02] worked on "fuzzy vault construct". In their theoretical contribution, the sender can place a secret S in a vault and locks (secures) it using an unordered set A (fingerprint minutiae was chosen in the solution). The receiver, using an unordered set B, can unlock the vault (access to S) only if B substantially overlaps with A. A and B are fuzzy templates.

Then, Uludag et al. [Uludag 06], based on the Juels and Sudan work, introduced a helper data by using template alignment to decrease the fuzziness criteria. They used polynomial interpolation to recover the secret key. In their experiment, the Genuine Acceptance Rate (GAR) was about 84,5% and False Acceptance Rate (FAR) equaled 0%.

The procedure to build the fuzzy vault (see Figure 3.7 replicated from [Uludag 06] page 165) is described as follows:

- Initially, Alice chooses a polynomial P of degree N which encodes S (S is generated as a 128-bit random bit stream, like an AES symmetric key. S is used as input parameter to construct P).
- Then, Alice calculates the polynomial projection, *P(A)*, where A is the reference minutiae points. (If *(x, y)* is an element of A, then u=x||y is used to calculate the value *P(u)*). "||" is the concatenation operator.
- Alice adds some points generated randomly (chaff points) whose images does not belong to P, to create the set of points R.

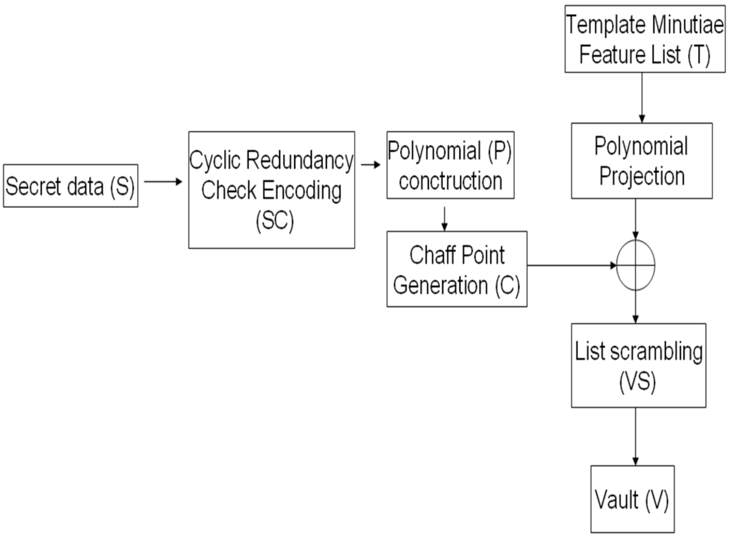

Template Minutiae Feature List (T)
Secret data (S)
Cyclic Redundancy Check Encoding (SC)
Polynomial (P) conctruction
Polynomial Projection
Chaff Point Generation (C)
List scrambling (VS)
Vault (V)

Figure 3.7: Fuzzy Vault Encoding

The Figure 3.8 ([Uludag 06] page 165) shows the decoding phase.

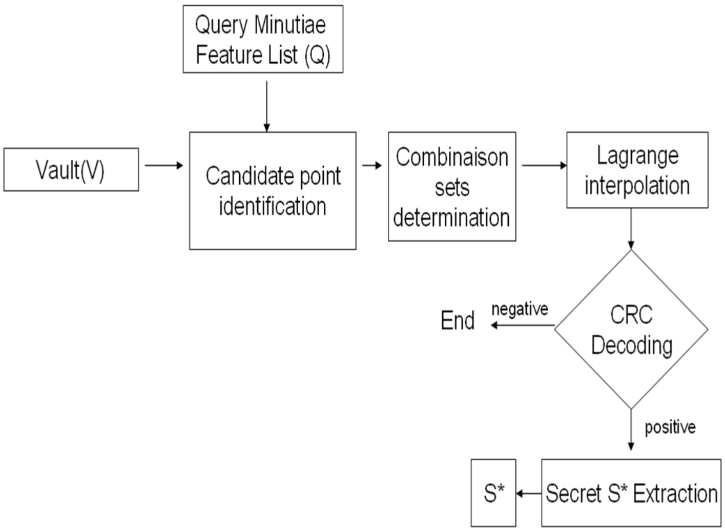

Query Minutiae Feature List (Q)
Vault(V)
Candidate point identification
Combinaison sets determination
Lagrange interpolation
End
negative
CRC Decoding
positive
S*
Secret S* Extraction

Figure 3.8: Fuzzy Vault Decoding

When Alice tries to recover S (by finding the coefficients of P), she uses her new enrolled minutiae points B. To decode the fuzzy vault, she needs to retrieve the same polynomial to extract the secret.

If B equalizes or differs a little from A, Alice is able to locate some abscissa u, which are used to interpolate P (their number must be equal to or higher than $N + 1$).

The algorithm decodes many candidate secret keys. Then, it finds which one of these candidates is the actual secret using Cyclic Redundancy Check (CRC).

The Fuzzy Vault concept using helper data was improved in [Nandakumar 07], and a new Palmprint Based Fuzzy Vault was developed by Kumar et al. [Kumar 09]. The performance of these systems is better than the old solutions and this is a proof of the fuzzy vault concept's usability to lock a key and share it between users.

3.5.1.3 Linnartz and Tuyls's Solution

Linnartz and Tuyls [Linnartz 03] assumed that there is always a perfect biometric template A (without noise) that is available in the enrollment phase. Those biometric features are aligned and the noise in each dimension is relatively small compared to quantification. Figure 3.9 shows the authentication system scheme.

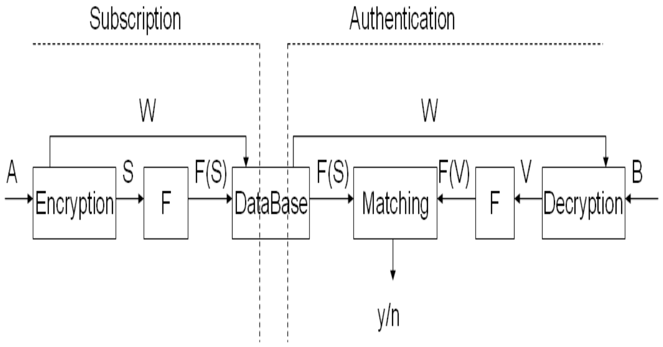


Figure 3.9: Authentication scheme

- In the *subscription phase*, Alice generates a secret S and a helper data W from a template A. She uses a hash function F to encrypt the secret S. *F(S)* and W are stored in a database. If the procedure is done offline, the verifier can obtain *F(S)* and W with a certificate.
- In the *authentication phase*, Alice enrolls a fuzzy template B. After recovering W from the database, the verifier creates V using B and W. He/She creates then *F(V)*. If $F(S) = F(V)$, the authentication is successful.

3.5.1.4 Dodis et al.'s Solution

Dodis et al. [Dodis 04] tried to turn the biometrical data into keys used in any cryptographic application. They proposed two functions:

- The first function is a *fuzzy extractor* that extracts nearly uniform randomness R from its biometric input; the extraction is error-tolerant since R is indifferent to small alterations in the input data. Thus, R can be defined as key for cryptographic use.
- The second function is a *secure sketch* which produces public information about its biometric input w that does not reveal w, but these information are a helper data to recover w even if the new enrolled value is close to w.

They also proposed some modification to the solution of Juels and Sudan [Juels 02] (fuzzy vault construction). Instead of adding chaff points with the point projection in the polynomial P, they proposed to employ a polynomial P'(of degree higher than P), which has common points with the points belonging to P. Thus, the new polynomial P' replaces the final set of points R.

3.5.1.5 Sahai and Waters's solution

The Fuzzy Identity-Based Encryption presented by Sahai and Waters [Sahai 05] used the Identity Based Encryption (IBE) scheme (described in the Section 2.7.2) with the biometric template as a source to generate public key. The error tolerance of a Fuzzy IBE scheme encourages the use of biometrical identities. Figure 3.10 shows a Fuzzy IBE scenario using iris as a biometrical data (this figure is replicated from the Sahai et al. presentation, page 19, available on line at:
userweb.cs.utexas.edu/~bwaters/presentations/files/Fuzzy-IBE.ppt).

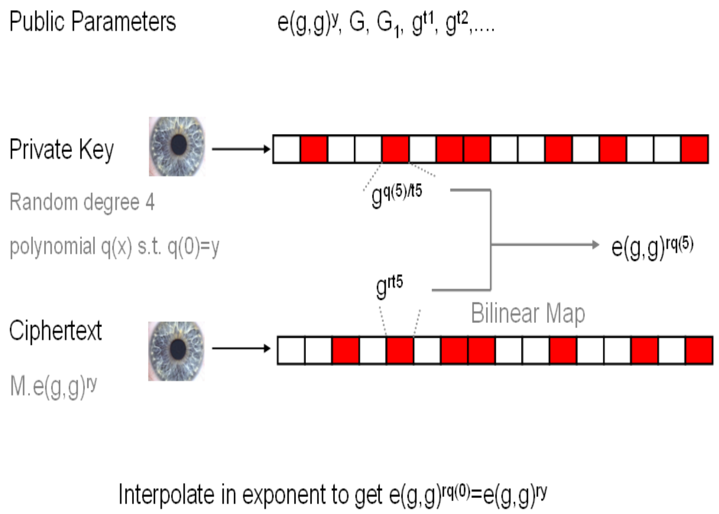

Public Parameters
$e(g,g)^y$, G, G_1, g^{t1}, g^{t2},....
Private Key
Random degree 4
polynomial q(x) s.t. q(0)=y
$g^{q(5)/t5}$
$e(g,g)^{rq(5)}$
g^{rt5}
Bilinear Map
Ciphertext
$M.e(g,g)^{ry}$
Interpolate in exponent to get $e(g,g)^{rq(0)}=e(g,g)^{ry}$

Figure 3.10: Fuzzy IBE

The system has a reference image A with well-defined characteristics. The authors choose a polynomial Q of order N and the secret is *Q(0)*. For each feature, they assign a coefficient of the polynomial Q. In the verification phase, they try to find at least $N+1$ features from B to interpolate the polynomial Q and thereby recover the secret.

In this solution, there is no relationship between biometrical data and the cryptographic key parameters. Each time there is the same iris characteristics in the first enrolled iris and the test iris, the coefficient of the polynomial at this position is used to retrieve *Q(0)*. We just remind that one of the goal of our work is to generate the cryptosystem parameters using the biometrical data which is not done in this solution.

3.5.1.6 Discussion

We present in Table 3.5 the list of the above discussed solution. We compare them by presenting the different methods, type of experiments and results in Table 3.5.

Techniques	**Experiments**	**Results**
Correlation Functions [Soutar 99]	theoretical	
Fuzzy Commitment [Juels 99]	theoretical	
Fuzzy Vault [Juels 02]	theoretical	
Helper Data [Linnartz 03]	theoretical	
Fuzzy Extractors [Dodis 04]	theoretical	
Fuzzy Identity Based Encryption [Sahai 05]	theoretical	
Fuzzy Vault & helper data [Uludag 06]	DB2 database of FVC 2002	FRR=15.5%/ FAR=0%
Fuzzy Vault & helper data [Nandakumar 07]	DB2 database of FVC 2002	FRR=3%/ FAR=0.24%
Fuzzy Vault [Kumar 09]	85 palmprints	FRR=0%/ FAR=0.4%

Table 3.5: Locking Key Solutions

We conclude that most of the discussed solutions are theoretical and that experimental performance values degrades comparing with the performances values given in Table 3.1.

3.5.2 Error Correcting Codes

With these solutions, researchers have tried to correct the characteristics of fuzzy biometrical data using Error Correcting Codes. The latter is an algorithm which detects and corrects any errors under some limitations, based on the remaining numbers in a set of symbols. The study of Error Correcting Codes and the associated mathematics is known as the coding theory.

3.5.2.1 Davida et al.'s Solution

Davida et al. [Davida 99] proposed an algorithm based on the iris code. They use multiple codes to achieve a canonical representation, which is associated with Error Correcting Code. They assume that multiple acquisitions of the iris are aligned.

In the *initialization phase*, the server puts in a smart card the following information:

– NAME: client name,

– ATTR: public attributes email address, ...,

– $\vec{C}$: Check digits for verification,

– Signature $Sig(Hash(NAME, ATTR, \vec{T}||\vec{C}))$ where Sig() is the signature of the authorized Server and Hash() is a hash function. $\vec{T}$ is the vector of K bits obtained upon enrollment. F

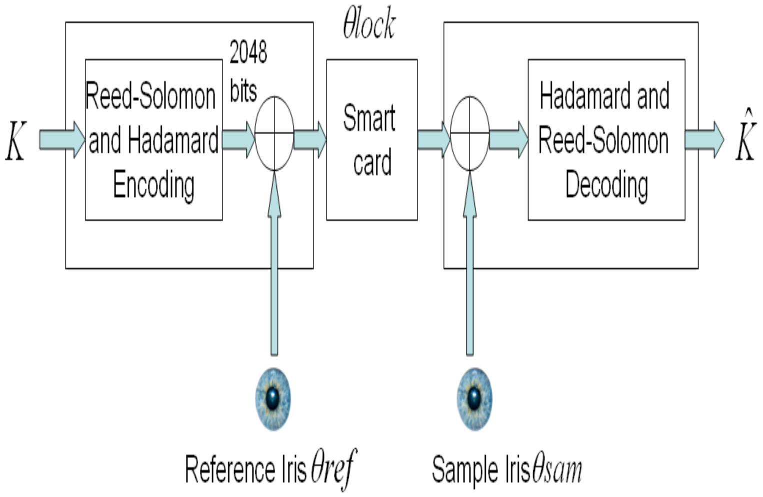
In the *verification phase* , a vector $\vec{T'}$ is obtained from the new enrolled biometrical data and the signature Sig(NAME, ATTR, $\vec{T'}$) is verified.

3.5.2.2 Hao et al. Solution

Hao et al. [Hao 06] proposed a new idea which aims at locking a 256-byte biometric key with iris code, after coding the secret. A well-known difficulty appeared which is resumed in 10 to 20% of error bits within the iris code. There are mainly two types of errors in iris codes: random errors caused by camera noise, iris distortion, etc., and burst errors generally resulting from eye-lids, eye-lashes, specular reflections, etc. This scheme uses Reed-Solomon codes [Reed 60] to cope with burst errors and Hadamard codes [Yarlagadda 96] for random errors. Intrinsically, the error correction capability of Hadamard codes is limited to up to 25%. But there are situations where iris codes can hold more than 25% variabilities.

To solve this problem, Hao et al. carefully studied the error patterns within iris codes. They used a two-layer error correction technique that combines both Hadamard and Reed-Solomon codes. They generated a key from a subject's iris image with the help of auxiliary error-correction data, which do not reveal the secret. The key can be saved in a tamper-resistant token such as a smart card. In their implementation, the corresponding False Rejection Rate (FRR) is only 0.47%.

A key K (random number) is generated. It is then coded using both Reed-Solomon and Hadamard codes to obtain a pseudo-iris code θ_{ps}. A reference θ_{ref} is chosen to encode the pseudo-code iris code and to get $\theta_{lock} = \theta_{ps} \oplus \theta_{ref}$. The result, to be put in token T, is $< K, \theta_{ref} > \Rightarrow T : \{\theta_{lock}, H(K)\}$. To retrieve the key, another sample of iris code is provided θ_{sam} so that $< \theta_{sam}, T > \Rightarrow \hat{K}$. The key is correct if and only if $H(\hat{K}) = H(K)$. The generated key is 140-bit long and the False Rejection Rate FRR is 0.47%. Figure 3.11 (replicated from [Hao 06] page 1083) shows the encoding and decoding of the secret key.



θlock
2048 bits
K
Reed-Solomon and Hadamard Encoding
Smart card
Hadamard and Reed-Solomon Decoding
K̂
Reference Iris θref
Sample Iris θsam

Figure 3.11: Iris Biometric Integration into Cryptographic Applications

3.5.2.3 Obtaining Stable Bit-String from Iris

It is well known that two biometric measurements are not completely identical as they contain some variability. Hence, it is not straightforward to use biometrics directly. A stable bit-string can be extracted from biometrics by using the cryptographic key regeneration system of Kanade et al. [Kanade 08] (which is based on the Hao et al. [Hao 06] scheme). It combines a randomly generated key with the iris data. This random key can later be extracted from the combined data by providing another genuine biometric sample. The regenerated secret can act as a cryptographic key. The key regeneration system is shown in Figure 3.12 (replicated from [Kanade 08] page 61).

Kanade et al. [Kanade 08] proposed a *zero insertion scheme* to reduce the error density in the iris codes thereby decreasing the number of errors per block. This increases the error correction capacity. The iris codes are shuffled using user specific *shuffling keys* to make them revocable. The shuffling also increase the separation between genuine and impostor Hamming distance distributions so that the verification performance of the system improves. The Reed-Solomon code can be used with various error correction capacity (t_s) settings. If this capacity is set to be high, amount of errors being corrected increases thereby decreasing the False Rejection Rate (FRR) and increasing the False Acceptance Rate (FAR). If the value of t_s is low, the FAR decreases but the FRR increases.

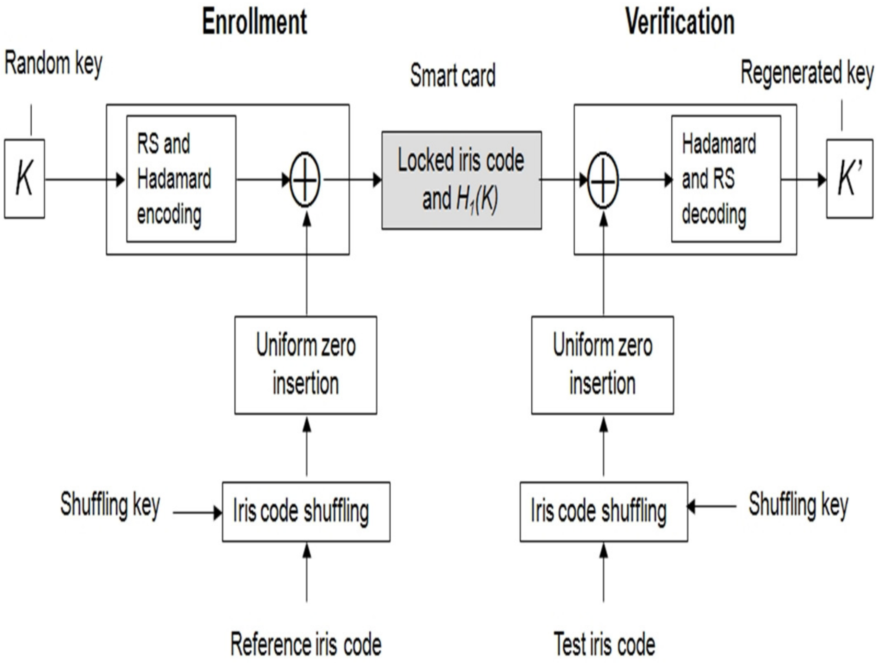


Figure 3.12: Cryptographic key regeneration using iris

Kanade et al. made their tests on the NIST-ICE database and they got better results than the Baseline biometric system (OSIRISv1) as shown in Table 3.6.

Techniques	Baseline biometric system (OSIRISv1)	Cancelable biometric system (i.e., with shuffling)
EER	1.71% (on ICE-Exp1)	0.23% (on ICE-Exp1)

Table 3.6: Results on NIST-ICE database

3.5.2.4 Discussion

In Table 3.7, we present the different discussed techniques and their experiment results. Since these solutions use the iris code, they are considered as the best biometric cryptosystems and they have a good performance results.

Techniques	**Experiments**	**Results**
Error Correction [Davida 99]	theoretical	
Reed-Solomon and Hadamard codes [Hao 06]	700 iris (private database)	FRR=0.47%/ FAR=0%
Shuffling key and error correcting codes [Kanade 08]	NIST-ICE database [NIST]	FRR=1.04%/ FAR=0.055%
Reed-Muller and product codes [Bringer 07]	NIST-ICE database	FRR=10^{-5}%/ FAR=5.62%

Table 3.7: Error Correcting Codes Solutions

3.5.3 Using Biometric Template as part of the Key

In this section, the biometric data is a part of the cryptographic key.

3.5.3.1 Monrose et al.'s Solution

Monrose et al. [Monrose 99], [Monrose 01] proposed a method to enhance security passwords using the keystroke dynamics and voice. In [Monrose 99], the goal was to utilize keystroke timings in the generation of a strong cryptographic key from a password. In [Monrose 01], a password is spoken by to generate his/her *derived key*, which would be the seed to a pseudorandom process to generate his/her private key. The latter would be used to decrypt incoming voice.

Their technique was inspired by password salting, where a random number named *salt* is added at the beginning of a password to make it hard to break. The keystroke dynamics and voices were used to produce the salts. The solution generates 60-bit cryptographic secrets.

3.5.3.2 Zero Knowledge Proof (ZKP)

Itakura and Tsujii [Itakura 05] use Zero Knowledge Proof (ZKP) to enhance the biometric cryptosystem. ZKP is an adaptation of the Schnorr identification and has the following steps:

P: Prover

V: Verifier

P's public key: $v = \alpha^{-s} \bmod p$, α an element of order q in $\mathbb{Z}/p\mathbb{Z}*$

P's private key: $s \in \mathbb{Z}/q\mathbb{Z}$

- P chooses a random digit $r \in \mathbb{Z}/q\mathbb{Z}$ and transmits $x = \alpha^r \bmod p$ to V.
- V sends a challenge $e \in [0, 2^k[$ to P.
- P sends $y = r + e.s \bmod q$ to V.
- V checks that $x = (\alpha^y v^e) \bmod p$.

The solution defined in [Itakura 05] proposes a multifactor biometric authentication method based on cryptosystem keys containing biometric signatures. These keys contain two kinds of data: personal biometrical data and a confidential random secret key. These keys are then the public keys in a Public Key infrastructure (PKI). Figure 3.13(replicated from [Itakura 05] page 291) shows the steps of the public key generation .

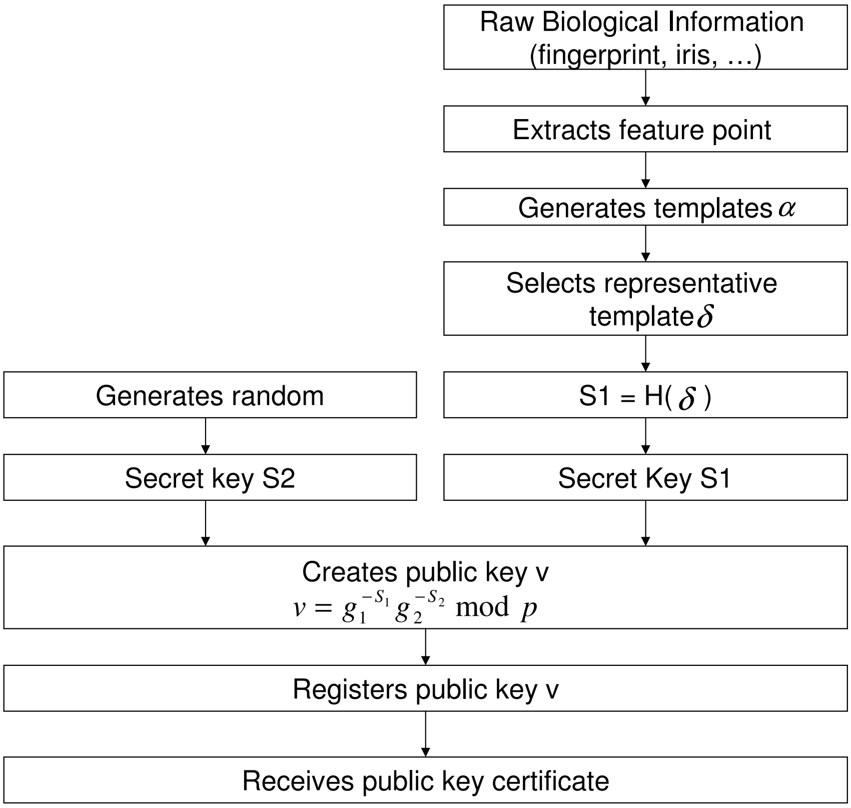

Raw Biological Information (fingerprint, iris, …)
Extracts feature point
Generates templates α
Selects representative template δ
Generates random
S1 = H(δ)
Secret key S2
Secret Key S1
Creates public key v
$v = g_1^{-S_1} g_2^{-S_2} \bmod p$
Registers public key v
Receives public key certificate

Figure 3.13: Public key generation

This is a description of the steps of Itakura and Tsujii solution:
P: Prover
V: Verifier
P's public key: $v = g_1^{-S_1} g_2^{-S_2}$ mod p, g_1 et g_2 an element of order q in $\mathbb{Z}/p\mathbb{Z}$*
$g_2 = g_1^{\alpha}$ mod p, (α is randomly chosen)
P's private key: S_1 (160-bit secret key based on biometrics)
and S_2 a traditional secret key with length between 160 bits and 1024 bits).
The 2 keys $\in \mathbb{Z}/q\mathbb{Z}$.
- P chooses 2 random digits r_1 and $r_2 \in \mathbb{Z}/q\mathbb{Z}$
and transmits $x = g_1^{r_1} g_2^{r_2}$ mod p to V.
- V sends the challenge $e \in [0, 2^k[$ to P.
- P sends $y_1 = r_1 + e.S_1$ mod q and $y_2 = r_2 + e.S_2$ mod q to V.
- V checks that $x = g_1^{y_1} g_2^{y_2} v^e$ mod p.

3.5.3.3 Burnett et al.'s Solution

Burnett et al. [Burnett 07] integrated the biometric template into an Identity Based Signature Scheme. The biometric template is embedded into a point on the elliptic curve. Then, it is used as part of the key pair generation for the signature scheme. The system was called Biometric Identity Based Signature Scheme. They used the solution given in [Dodis 04] to generate key data from Biometrics.

3.5.3.4 Discussion

In Table 3.8, we present a summary of the discussed solutions. Most of the works are theoretical and the solutions proposed by Monrose et al. need to be improved.

Techniques	**Experiments**	**Results**

Salting procedure [Monrose 99]
[Monrose 01]
	keystroke dynamics (13 persons)
	voice recording
		FRR=48.4%
		FRR=6%

Zero Knowledge Proof [Itakura 05]	theoretical	
Identity Based Signature Scheme [Burnett 07]	theoretical theoretical	

Table 3.8: Biometric Template as part of the Key

3.5.4 Authentication, Authorization, Accounting (AAA) and Biometry

In this section, we describe the combination of Biometry with Authentication, Authorization, and Accounting (AAA) protocol. The latter performs three functions: authentication, authorization and traceability. Extensible Authentication Protocol (EAP) [Aboba 04] is an authentication mechanism, frequently used in wireless networks and connections Point-to-Point Protocol (PPP). It is generally used with AAA protocols like RADIUS [Rigney 00] or DIAMETER [Calhoun 03].

One idea was to use the biometric template as an identifier of the person in EAP messages exchange. SentriNET [Bersani 04] is designed to expand access to distance through a Network Access Server (NAS). Its basic operation involves sending a username to the server. The latter verifies the type of biometrical data and returns relevant information to the user's sensor . The client captures the template and returns it to the server for verification.

Someone may say that the biometric data plays the role of a cryptographic key but in reality, there is a private key that is associated with each user. So there is no EAP solution directly using a key-dependent biometric template. Hence, the biometrical data are used to identify the client and not to authenticate him/her.

There is another solution to use the biometry in an Extensible Authentication Protocol (EAP) authentication mechanism. This solution was proposed by Lee et al. [Lee 06], and it was applied to Home Network (HN). This approach is illustrated in Figure 3.14(replicated from [Lee 06] page 3). A new message is added to EAP Authentication Response, so the mechanism is similar to Tunneled TLS "Transport Layer Security" (EAP-TTLS) [Funk 06].

The required steps of this solution are as follows (The server is a Home Gateway (HG) in this scenario): In the *initialization phase*, user's Biometric Templates (BTs) are enrolled at the Home Gateway (HG) (stored in a database).

In the *authentication phase*:

1. User's device (i.e; user's authentication client module) authenticates the HG by verifying its certificate.
2. The user and the HG then share the same key and cipher suites through using TLS protocol.
3. The user's Biometric Template encrypted by the shared key is transferred to the HG.

4. The user's authentication result is transferred to the user device in an encrypted form, where legitimate users get authenticated.

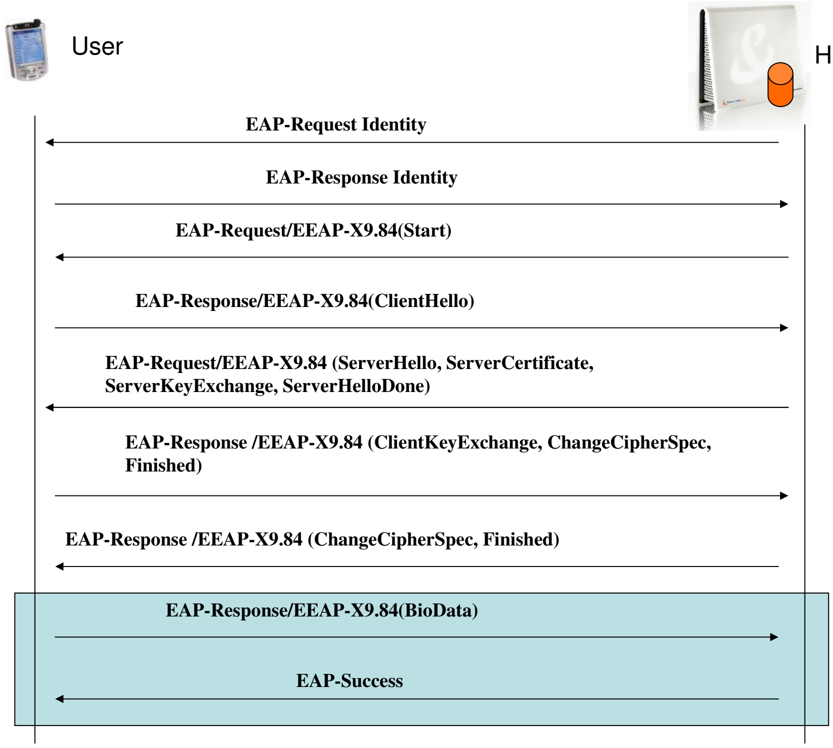

User
Home Gateway
EAP-Request Identity
EAP-Response Identity
EAP-Request/EEAP-X9.84(Start)
EAP-Response/EEAP-X9.84(ClientHello)
EAP-Request/EEAP-X9.84 (ServerHello, ServerCertificate, ServerKeyExchange, ServerHelloDone)
EAP-Response /EEAP-X9.84 (ClientKeyExchange, ChangeCipherSpec, Finished)
EAP-Response /EEAP-X9.84 (ChangeCipherSpec, Finished)
EAP-Response/EEAP-X9.84(BioData)
EAP-Success

Figure 3.14: EAP method using Biometrical Data as Identity

3.6 Conclusion

The biometrics deployment continues to spread in many fields. Despite that, biometrics still faces fraud and identity theft. Cryptography, with its conventional methods, is a good way to make biometrics more trusted within international organizations.

We presented in this chapter the biometric cryptosystems which aim to combine biometry and cryptography to strengthen the biometrical identification procedures. We classified these cryptosystems into 3 types: locking the key using biometric template, using Error Correcting Codes and using Biometric Template as part of the key. We described some contributions for each types and we noted that in general, the proposals are theoretical and few experimental tests are done. We presented also the Authentication, Authorization, Accounting (AAA) and Biometry combination as a new method to authenticate a user.

The future solutions must allow the use of personal biometrical data while leaving information in the human body. Also, the use of elliptic curve Cryptography is one of the best choice to improve existing solutions. In Chapter 4, we propose a solution that uses cancealable biometric, locking key method and the AAA mechanisms. In chapter 5, we propose two solution based on Elliptic Key Cyptography and Error Correcting Code.

Chapter 4

Home Network (HN) Case Study

4.1 Introduction

As more homes are outfitted with computers and personal devices (for example, Personal Digital Assistants (PDA)s, cellular phones, MP3 players, game consoles), it is natural to think of interconnecting devices for data and access sharing peripherals (such as printers). As a consequence, users are willing to explore the possibility of connecting the Home Network (HN) to the Internet. In this chapter, we focus on the authentication of each user within the HN, based on his/her biometrical data to access the local services and the Internet. This was a preliminary work to study the usability of biometrical data to strengthen authentication mechanisms in the Home Network (HN).

As defined in [Lee 06], a HN is composed of personal devices, at home, connected to a local gateway, also known as the Home Gateway (HG) or the Home Server. In fact, the HG plays the role of a communication-gateway between the indoor and the outdoor world (i.e; between home and the Internet). In the context of our work, HG also plays the role of an authentication server, while users' devices (such as PDAs, laptops, PCs...) are considered as authenticated clients.

In [Ellison 02], Ellison defines many kinds of HNs based on home users. These are as follows:

i) Single-Person homes,

ii) Couple with Small Children,

iii) Families with Teenagers, and

iv) Adult Guests and Roommates.

In our study, we consider the last two types, since they need more constraints in the security policy especially concerning the access to the Internet and to the possible services in one hand, and they present an important market to telecom operators and service providers on the other hand. In fact, Biometric authentication is promising in the HN scenario.

The rest of this chapter is organized as follows: Section 4.2 discusses some related works on Home Network Access; Section 4.3 presents the new proposed solution; Section 4.4 presents an analysis of the proposed solution. Finally, the chapter concludes in Section 4.5, highlighting some future work.

4.2 Related Works

This section presents some existing solutions for users' authentication in HNs. We consider that most devices in HNs can use wireless connection (IEEE 802.11) with a Home Gateway (HG).

The first solution [Borisov 01] was based on Passwords. Since HNs' users do not always have very strong knowledge on Network Security, they usually choose weak secrets or they forget to change it in a periodic manner, ignoring the necessity to do this. Another solution consists in using certificate authentication methods [Kim 09], which is more secure but has the problem of certificate distribution and revocation.

Lee et al. [Lee 06] also proposed the solution described in Section 3.5.4. Biometric authentication is applied there. However, some limitations exist. We noticed in this solution that the Biometric Template (BT) is just encrypted and then sent in the network which makes the personal data exposed to attacks. This solution is called Encrypted Extensible Authentication Protocol (EEAP). Thus, the solution is less compatible with existing Extensible Authentication Protocol (EAP) protocols and hence is difficult to be deployed at a commercial level. Moreover, it is not clear how the same key is exchanged between the client and the server to encrypt the BT. Also, conforming to the CNIL recommendation on the storage of BT by non governmental authorities, the proposed solution assures a local storage of the acquired fingerprints.

A new solution is provided by "Windows 7", known as the Windows Biometric Framework (WBF) [Microsoft 10] that permits users to login to their windows account using their biometrical data (WBF supports only fingerprint biometric devices). In Windows 7, there is a common management platform for different fingerprint biometric devices. Therefore, WBF is a generic and common platform for different vendors. WBF is used in two primary end-to-end scenarios:

- Logon: Users use their fingerprint to log on to a local machine or to a domain.
- User Account Control (UAC): A system administrator can elevate applications using a fingerprint.

We noticed that the above discussed solutions, except Lee et al. and Windows Biometric Framework (WBF), mainly aim at authenticating the Home Network (HN) itself regardless which user is being connected (i.e; the network operator/service provider only identifies the HN owner "subscriber"). But at a finer granularity level, each user is a separate entity having his/her profile, thus separately authenticated using personalized security parameters. At the same time, the privacy of each user should be guaranteed. In this context, biometric authentication is a promising solution allowing identification of each user according to his/her BT and thus authenticating him/her in a distinguished manner and personalizing his/her access.

In this chapter, a modified biometric authentication mechanism is presented where BT is not transmitted through the air. The proposed solution is detailed in the next section and analyzed in Section 4.4.

4.3 Our proposal: Personalized User's Access in HNs

This section presents a new solution, in which another level of security is added through fine granularity authentication. It aims at personalizing the access of each user in the HN using their biometrical data and preventing illegitimate users (passing by the HG) to have access to any devices. Also, we need a local storage of biometrical data conforming to the CNIL recommendation on the storage of Biometric Template (BT) by non governmental authorities.

Our approach of personalized access allows each user to use any device in the HN, while being able to access his/her appropriate profile. We propose a new biometric authentication method, while keeping in mind that the HG does not store any users' Biometric Template (BT). To satisfy this constraint, we propose using the Fuzzy vault method (see Section 3.5.1.2) to hide a secret that should be used for authentication. The HG has the role of generating a secret for each user for each session which is hidden by the BT. The user needs to recover the secret in order to be authenticated.

4.3.1 Hypothesis

We are concerned with several questions about the feasibility of the solution and we consider different hypothesis like:

– *Hypothesis 1*: The fingerprint acquisition of each user is carried out using the HN equipments, which are supposed to have integrated biometric sensors. We think that these hypothesis is feasible since many equipments are equipped with fingerprint sensors (like laptop, PDA,...). This solution is one of many proposed to the HN's users, if they are suspicious about using this technology, the basic solution of password is still available.
– *Hypothesis 2*: Each user should enroll his/her fingerprint using the fingerprint sensor embedded in the equipment. Then, a Biometric IDentifier (BioID) is generated and stored in the equipment and the Home Gateway (HG). This procedure is done manually because there is no secure wireless session in the configuration phase. There is a problem related to the guests which are temporary users of the system. One solution is to let them enroll their fingerprint to start a new session. But, we think that this solution is infeasible since it is an embarrassing method for them. The best solution is to create a default guest account without biometric recognition.
– *Hypothesis 3*: The BioIDs for each member of the family are stored in the equipments. This means that each member need to have an account in these equipments. This is a preventive method to prohibit the use of some equipments or to access forbidden services (like for children).
– *Hypothesis 4*: The BioID's generation is based on Cancelable Biometrics described in Section 3.4. This measure can prevent the loss of biometric data when the equipments or the Home Gateway (HG) are stolen or lost. Using this method helps to overcome the non-revocability of biometrics.

4.3.2 Description of the Solution

We consider a Home Network (HN) scenario in which users connect to a Home Gateway (HG) for broadband Internet access, using any equipment in the HN. Figure 4.1 illustrates the context of the proposed solution.

Each user should enroll his/her biometric template (BT) to be authenticated. The objective is to allow each user in the HN to have a personalized access and to access his/her proper personal context. From an operator point of view, the proposed solution respects the operational constraints as well as the constraints posed by the CNIL (French

Data protection Authority) [CNIL 07] concerning the biometry's use. The operational requirements concern the compatibility of the proposed solution with the Authentication, Authorization and Accounting (AAA) architecture at the operator's network, where there is no need to use new authentication protocols or modify existing ones. While, the CNIL's requirements concern the illegal storage of BT as well as its non revealment.

Assume that a Biometric Identifier (BioID) is created using user's BTs, and stored locally in the HN (limited to the HG and the HN equipments). This identifier is not transferred in the network. One should also notice that the size of the storage space is not huge (limited to the number of family members at home).

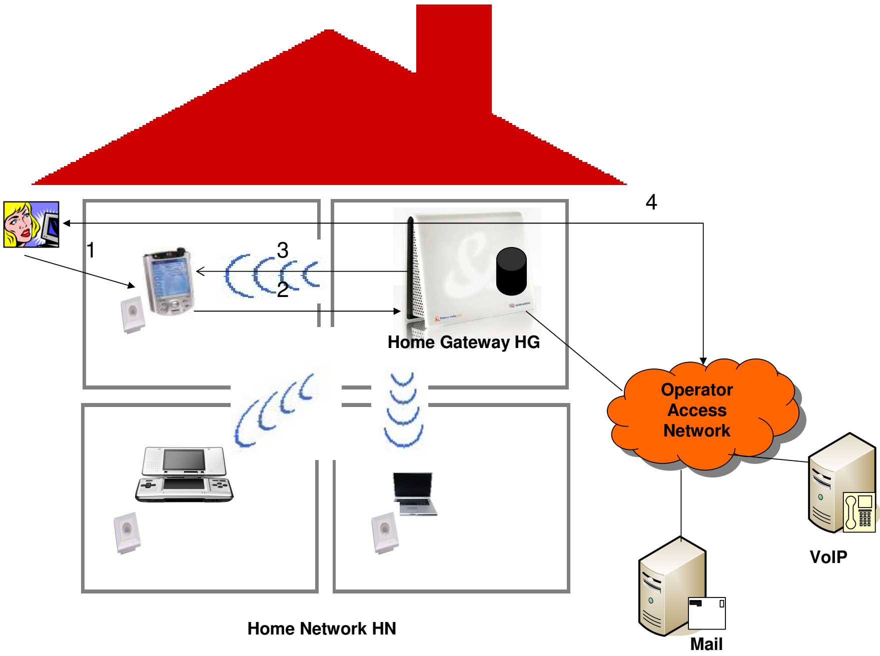


Figure 4.1: Access Personalization in Home Networks HN

4.3.3 Required Conceptual Phases

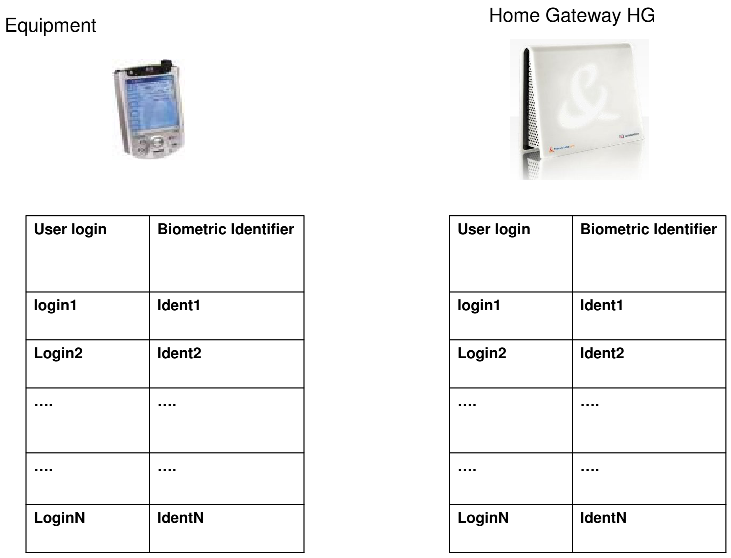
The proposed solution requires three phases, which mainly concern the equipments configuration, the Biometric Template (BT) treatment and storage.

4.3.3.1 Configuration Phase

In this phase, each user should present his/her BT (fingerprint template) to be manipulated and stored in the database of the HG.

The digital fingerprint is enrolled using a biometric sensor at the HN equipment. Then, a software, generates the BioID using the functional transformation described in Ratha et al.'s paper [Ratha 07]. The Figure 3.4 presents the output which is a modified set of minutia points. The latter can not be used to revoke the original fingerprint. A predefined number of these minutiae is then selected to create the BioID. For example, one can choose to use 24 minutiae points to create a 384-bit BioID.

Users' BioIDs are then stored in a table form (this could be a special file) together with the logins that correspond to their owners. Figure 4.2 shows the storage form of users' BioID in the HN equipment and in the HG.



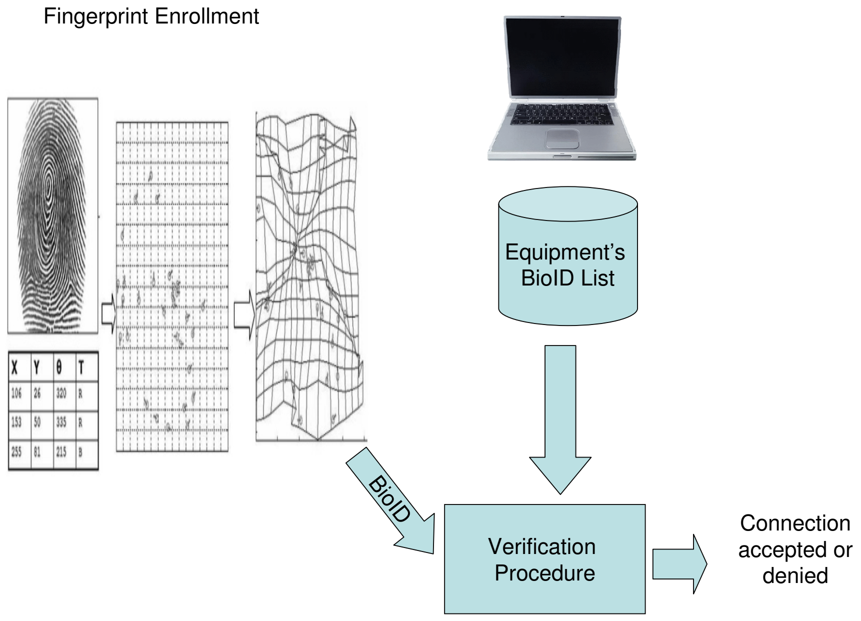
User login	Biometric Identifier
login1	Ident1
Login2	Ident2
....	....
....	....
LoginN	IdentN

User login	Biometric Identifier
login1	Ident1
Login2	Ident2
....	....
....	....
LoginN	IdentN

Figure 4.2: BioID Storage in the Home Network HN

4.3.3.2 Users' Connection to the Home Gateway

Each time the user wishes to connect to the HG, he/she does a new acquisition for his/her fingerprint in order to identify him/herself, without any need to type a login or a password. In this case, the fingerprint acquisition is done through the HN equipment that is being used, and they are treated (as explained in the previous phase) to generate the user identifier BioID. The generated identifier is then compared with the one stored in the equipment for the same user. If the same identifier exists, the corresponding login is sent to the HG and the process of user's authentication starts. Figure 4.3 presents the procedure to connect a user to the Home Network (HN).



X	Y	θ	T
106	26	320	R
153	50	335	R
255	81	215	B

Figure 4.3: User's Connection Procedure

4.3.3.3 Users' Biometric Authentication

When the HG receives the user's login, it searches in its database for the corresponding BioID to this user. Then, it starts authenticating the user based on this identifier in order to allow him/her to have personalized access. The authentication process is mainly based on a challenge-request/challenge-response approach. The mechanism is a modified version of the Extensible Authentication Protocol (EAP) and it focuses only on the exchange between the user and the HG. There is no contact with the operator's server. In fact, only the vault and the challenge need to be piggybacked in the EAP-request. This highlights that this solution is open for any EAP method.

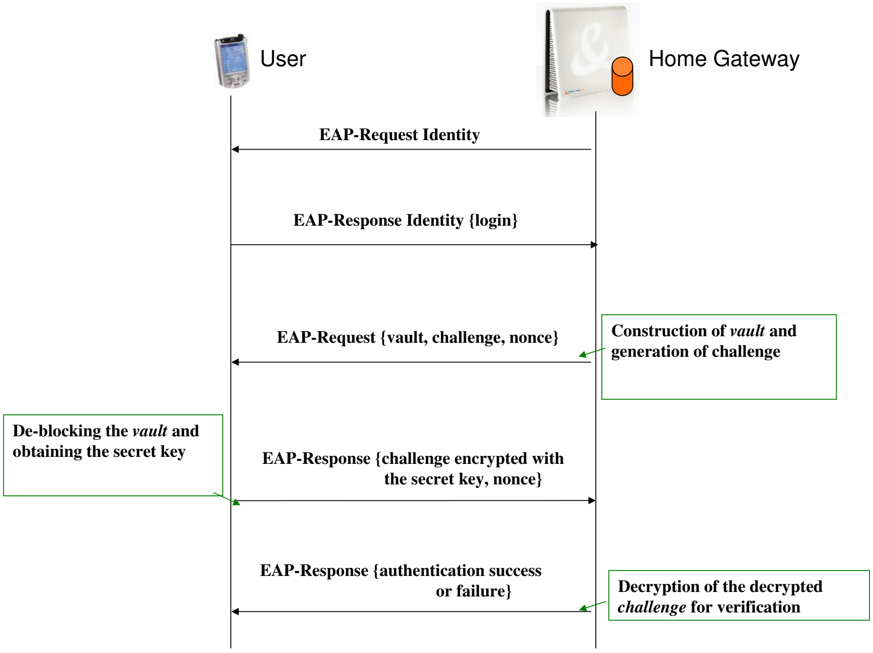

User
Home Gateway
EAP-Request Identity
EAP-Response Identity {login}
EAP-Request {vault, challenge, nonce}
Construction of *vault* and generation of challenge
De-blocking the *vault* and obtaining the secret key
EAP-Response {challenge encrypted with the secret key, nonce}
EAP-Response {authentication success or failure}
Decryption of the decrypted *challenge* for verification

Figure 4.4: Messages Exchange for Authorizing the Personalized Access

Figure 4.4 illustrates the corresponding messages' exchange:

- A secret key is generated by the HG. The latter chooses a challenge for the user that wishes to connect.
- The Fuzzy Vault [Uludag 06] method is applied in order to construct a vault that hide the secret key. The BioID is used to create the vault following the method described in Section 3.5.1.2.
- The resulting vault and the challenge are transmitted to the user. The HG adds a nonce to prevent any replay attack.
- The user unlocks the vault using his/her BioID (resulting from the current fingerprint enrollment) in order to retrieve the secret key.
- Once the secret key is found, the user transmits the encrypted challenge with the recovered secret key to the HG. He/She adds the nonce to the message.
- The HG decrypts the challenge using the secret key and compares it with the one initially sent. If they match, the user is authenticated and he/she gets a personalized access.

The message's details are given in Appendix B.

4.4 Solution Analysis

This section presents practical considerations for the deployment of our solution and it's security analysis.

4.4.1 Practical Considerations for the Solution Deployment

The proposed solution has many advantages from the user point of view. First, it allows an easy way to access services (each user simply presents his/her fingerprint). Then, it ensures personalized users' access in spite of the equipment/terminal that is being used. Also, conforming to the CNIL recommendation on the storage of BT by non governmental authorities, the proposed solution ensures a local storage of acquired fingerprints. Additionally, the fingerprints are treated before being used or stored. This allows decreasing the risk of their theft in case the HG is compromised by an intruder for instance.

Moreover, from the network operator/service provider point of view, the proposed solution is compatible with the existing Authentication Authorization Accounting (AAA) infrastructure. Only the vault and the challenge in one EAP-request need to be piggybacked. In the end, this solution is promising in opening new business opportunities, thanks to the biometric authentication method that allows for a personalized users' access and hence a better access control in HNs. For instance, it makes it easier to monitor and control the children's access to the Internet, even when their parents are away. The personalized users' access in the proposed solution allows controlling the access for each broadband access line depending on the user that is being connected. However, in classical broadband access control, the connection itself is authenticated. This is considered as a part of the configuration phase. The authentication takes place each time the Home Gateway (HG) is granted an IP connectivity (and hence the Internet connection for a user).

4.4.2 Security Consideration

The proposed solution prevents the storage of users' Biometric Templates (BTs) in operator's databases. Only some random fake minutiae should be stored, which represent the Biometric IDentifier (BioID). The latter should be enough to identify the fingerprint's owner but insufficient to recover the whole fingerprint. These fake minutiae are generated using the functional transformation described in Ratha et al.'s paper [Ratha 07]. Changing the secret key each time the user connects should avoid attackers from retrieving the BioID.

4.4.2.1 Attacks on the Home Gateway (HG)

During the communication with the HG, we assume that the HG uses its certificate (previously obtained by a Certificate Authority (CA)). Therefore an attacker (an illegitimate user) could not decrypt the communication between the HN's users and the HG, since he/she needs the HG's private key to decrypt the message encrypted with the HG's public key. On the other hand, if an attacker wants to impersonate the HG, he/she could not generate a valid vault, since he/she does not have the BioID.

4.4.2.2 Impersonating Users

We found in the literature that the fuzzy vault mechanism with helper data [Uludag 06] has False Rejection Rate (FRR) equals 14.5% and False Acceptance Rate (FAR) equals 0%. In the [Nandakumar 07] experiments, the FRR equals 3% and the FAR equals 0.24%. As we mentioned in Table 3.5, these performance values are the result of experiments on DB2 database of FVC 2002. The database can be downloded from the website of the Second International Competition for Fingerprint Verification Algorithms (http://bias.csr.unibo.it/fvc2002/). These results demonstrate the good choice of the fuzzy valut mechanism in our solution. When a malicious user wishes to impersonate the legitimate user using his/her own fingerprint (i.e; different BioID), he/she is neither able to decode the vault nor able to encrypt the challenge with the secret key.

4.5 Conclusion

Home Networks (HNs) security is an emerging research field, attracting both the research community and the industry. An important trend is to separate user's authentication from the used devices, allowing for fine granularity authentication and users' personalized access in spite of the devices' authentication in the HN. Our solution answers CNIL's requirements about local storage of biometrical data and reinforced protection of this sensitive data.

Applying biometric authentication is promising in allowing users' authentication in a distinguished manner as well as personalized users' access. However, this technology should be carefully used in order to protect users' privacy and prevent the disclosure of their Biometric Template (BT). Our proposed solution allows the protection of private BT thanks to applying the fuzzy vault mechanism. We do not only propose an algorithm but also all the procedures that go along to provide a simple but still very robust solution.

In chapter 5, we consider another environment which is the Government Network. We continue to enhance the authentication mechanisms by deploying biometrics.

Chapter 5

Government Network Case Study

5.1 Introduction

Since 2004, many countries all over the world, have released electronic Passports (e-Passports) containing biometric data. The evolution of cryptographic protocols for e-Passports has led to their widespread deployment [Nithyanand 09]. These e-Passports have an embedded Radio Frequency Identification (RFID) chip which is capable of cryptographic functionality like Elliptic Curve Cryptography (ECC) [Batina 06].

The introduction of biometrics and the implementation of RFID technologies in the e-Passports aim to strengthen border control by reducing falsification and establishing reliable identification of the document's bearer.

After the adaptation of the International Civil Aviation Organization (ICAO) standard [ICAO 06] in many countries such as the USA, there was evidence of inadequate data protection and weaknesses in the privacy protection [Juels 05]. To improve the e-passport security, the European Union (EU) has released a new specification which includes a set of protocols called Extended Access Control (EAC) [EU 06]. The new protocol solves some of the previously cited problems. The EAC protocol performs a mutual authentication between the RFID chip embedded in the e-Passport and the RFID reader.

After the introduction of EAC, some researchers tried to improve theses solutions by solving the problems such as the certificate revocation and the use of a session key with insufficient entropy. They proposed an Online Secure e-Passport Protocol (OSEP) [Pasupathinathan 08b]. This protocol is based on the existing ICAO PKI implementation (first generation e-Passports) but eliminates the cross certification, between participating countries, needed in the EU-EAC (second generation e-Passports).

In this chapter, we present and analyze the previous security protocols in e-Passport. We, then, present our proposal which applies Elliptic Curve Diffie-Hellman (ECDH) Key Agreement to generate a session key. The elliptic curve's parameters used in the biometric cryptosystem are generated by using the minutiae data (fingerprint) of the e-Passport's bearer.

Then, we present an iris based authentication mechanism for e-Passport. It is a modified version of the first scheme and it uses iris code instead of fingerprint. Kanade et al. [Kanade 08] scheme is employed to obtain a key from iris biometrics and this key is used to generate the security parameters.

This chapter is structured as follows. Section 5.2 discusses the technical features of the e-Passports. Section 5.3 presents the ICAO, the EAP and the OSEP solutions. Section 5.4 presents our first solution which uses fingerprint as biometrical data to reinforce security in e-Passport protocol. Section 5.5 presents the modified solution which uses iris code. Section 5.6 discusses a security analysis of our solution, and Section 5.7 present the implementation and the performance evaluation of the new mechanism. Section 5.8 concludes this chapter.

5.2 Technical Features of e-Passport

In this section, we present the technicals features of the e-passport.

5.2.1 Structure of Machine Readable Zone (MRZ)

The ICAO developed standards for Machine Readable Travel Documents (MRTDs) [ICAO 03], including passports and visas, with the intention of speeding up the passport's control procedure at border crossings. Every MRTD possesses a special Machine Readable Zone (MRZ). The MRZ is composed of two lines and each line has 44 characters. The following information is provided in the passport's MRZ: name, sex, date of birth, nationality, passport number, date of expiry and check digits. They passport's number, date of birth and date of expiry are essential elements for the e-Passport security.

In the IBM report research [Kc 05], the authors presented the next generation of MRZ which is 2-D barcodes. The latter encode approximately 8192 bytes of information and are currently applied in many passports, visas, and driving licenses".

5.2.2 Data Structure of the e-Passport

As already mentioned, an RFID chip is embedded in each electronic passport (e-Passport). A file system for the chip is defined as Logical Data Structure (LDS) which is specified in a technical report [ICAO 06]. The LDS initially consists of 16 data groups. In the future, there will be three additional data groups such as visa of the destination country or travel record details. The default mandatory biometrical data to be stored is the traveler's headshot. The fingerprints and iris images are optional. In Table 5.1, we present the data embedded in the contactless chip conforming to the ICAO guidelines.

Data Group	**Data Element**
DG 1	Document Details
DG 2	Encoded Headshot
DG 3	Encoded Fingerprint
DG 4	Encoded Iris
DG 5	Displayed Portrait
DG 6	Reserved for Future Use
DG 7	Signature
DG 8 - 10	Data Features
DG 11 -13	Additional Details
DG 14	CA Public Key
DG 15	AA Public Key
DG 16	Persons to Notify
SOD	Security Data Element (SDE)

Table 5.1: E-Passport Logical Data Structure

DG: Data Group, SOD: Document Security Object

5.2.3 The e-Passport Public Key Infrastructure

A Public Key Infrastructure (PKI) is needed to perform the process of public key distribution and authentication. The entities interacting within the e-Passport PKI are the Country Verifying Certificate Authorities (CVCA) also known as Country Signing Certificate Authorities (CSCA), the Document Verifiers (DV), and the Inspection Systems (IS).

The Public Key Infrastructure usually has a hierarchical structure. The CVCA is in the top level in each country. It generates and stores a public/private key pair (PK_{CVCA},

PrK_{CVCA}). The private key of the CVCA (PrK_{CVCA}). Each Document Verifier (DV)'s certificate is signed using the CVCA's private key (PrK_{CVCA}). The DVs can be of the same country or from other countries. Usually in each country, there are many Document Verifiers. Each of these Document Verifiers generates and stores a public/private key pair (PK_{DV}, PrK_{DV}). Each Inspection System (IS)'s certificate is signed using the DV's private key (PrK_{DV}). Also, The DV signs the Security Data Element (SDE) of the e-Passports it delivers.

The DVs' certificates, of all countries, need to be shared. Therefore, the ICAO provides a Public Key Directory (PKD). The PKD only stores the certificates of all registered DVs. This list of certificates is public and used by all countries. All certificates have a limited validity period. The PKD can store different Certificate Revocation Lists (CRL). Every country is responsible for updating its own storage of public certificates and CRL's. After downloading them from the PKD, each country spreads the newly downloaded information to every Document Verifier (DV) and Inspection System (IS) in its authority.

5.3 Security Protocol in e-Passport

It is important to note that e-Passport makes travelers confused [Vaudenay 07]; they ask if it is secure to leave their biometric templates (face, iris, and fingerprint) in chips which are exposed to clandestine scanning.

In the e-Passport, the biometric templates stored in the chip are the face figure, the fingerprint minutiae and the encoded eye (iris). We note that the US-VISIT program [US-VISIT 04] requires fingerprint biometrics from visitors. Since January 2003, border control officers have been recording facial images and index fingerprint images for visa carrying passengers upon arrival at a US border control posts.

Since the biometrical data are sensitive and they are stored in a contacless chip, we need to focus on the vulnerability of such technology. Implanting a contactless chip in the e-Passport offers several advantages comparing to contact smart chips:

- no wear and tear due to frequent usage
- data transmission rates is faster
- no need to change the e-Passport's cover by adding contact chip.

Nevertheless, contactless chips have two major drawbacks. In the first place, as the transmission is done in wireless manner, the surrounding readers (not the legitimate receiver) may collect information. In the second place, a reader finds difficulties to sort the transmission coming from a particular chip (if there is many contactless chips very close in reader's entourage).

The following sections present the different solutions found in the literature to overcome these weaknesses.

5.3.1 ICAO First Generation e-Passport Specifications

The International Civil Aviation Organization (ICAO) specifies some cryptographic measures to ensure authenticity and privacy of biometric data [Juels 05]. The New Technologies Working Group (NTWG) works on the specification of smart card based biometric passports.

In 2002, the U.S. Congress passed the Enhanced Border Security and Visa Entry Reform Act. 27 US Visa Waiver Program (VWP) nations projected to issue e-Passports that are resistant to tampering. The e-Passports would incorporate biometrical data and document authentication identifiers like described in the ICAO's standard. Mid-2005 was the deadline given by the USA to the countries to produce, procure and implement the e-Passport in practice.

There are three cryptographic protocols described in the first generation ICAO's specification to ensure data correctness and privacy. They are: Passive Authentication (PA), Basic Access Control (BAC), and Active Authentication (AA).

5.3.2 Passive Authentication (PA)

Passive Authentication (PA) is the only *mandatory* cryptographic protocol in the ICAO first generation specification. Its primary goal is to allow the Inspection System (IS) to check if the e-Passport's data is authentic and the data's integrity is preserved. But, it does not confirm the authenticity of the chip itself (it can not detect cloning). The IS obtains the issuing Document Verifier (DV)'s certificate. It needs the DV's public key to verify the digital signature of the data in the LDS. After the signature's validation, the Inspection System (IS) hashes each one of the Data Group (DG) and compares the hashes with the values stored in the SOD. If there is a match, the data on the chip is not falsified.

5.3.2.1 Active Authentication (AA)

Active Authentication (AA) is an *optional* protocol in the ICAO's specifications. Using a simple challenge/response mechanism, the algorithm detects if a Chip has been substituted or cloned. If Active Authentication (AA) is supported, the Data Group 15 is the AA public key and its hash value is stored in the SOD. The corresponding private key is stored in the secure section of Chip memory. To prove its authenticity to the Inspection System (IS), the chip must convince the IS that it possesses this private key.

5.3.2.2 Basic Access Control (BAC)

Basic Access Control (BAC) is an *optional* protocol that ensure that only the Inspection System (IS) can read the e-Passport Chip's data. The IS starts a challenge/response protocol to prove to the passport that it has read optically the contents of the machine readable zone (MRZ). This data is used to generate the key seed K_{seed}. The key seed K_{seed} is derived from the following MRZ's data: The e-Passport Number (PN), Date of Birth of the Passport's bearer (DOB), Date of Expiry (DOE), 3 Check Digits (C).

$$K_{seed} = 128msb(SHA - 1(PN||DOB||DOE||C))$$

(where 128msb : 128 most significant bits and SHA-1: hash function)

Using K_{seed}, the IS and the chip compute a key for Message Authentication Code (MAC) and a session key to provide confidentiality and integrity for any communication between them. A 32-bit sequence counter is included to prevent messages replay. The keys are fresh for each session. Hence, The Basic Access Control (BAC) prevents hostile reading problem (called skimming) of passports. We notice that BAC does not authenticate the Inspection System (IS): anyone who optically read the MRZ can successfully complete BAC and access the chip's storage memory.

5.3.2.3 Weaknesses

In 2005, Ari Juels et al. [Juels 05] described the privacy and security issues of the ICAO specifications and then, gave some solutions to some of the existing weaknesses. In 2008, Pasupathinathan et al. [Pasupathinathan 08a] provided a formal security analysis for ICAO e-Passport implementation (in fact, Australian one). They concluded that ICAO e-Passport guideline had some weaknesses that are listed in below:

– The e-Passport's protocols do not assure data origin authentication as it do not prevent replay and grandmaster chess attacks (the latter is to have a fake e-Passport between the IS'reader and the legitimate chip).
– Data confidentiality is not satisfied because an attacker, after reading the data in Machine Readable Zone (MRZ) is able to obtain encryption and MAC keys stored in the e-Passport's chip. Thus, the security goals for Active Authentication protocol, like mutual authentication, key freshness and key integrity can be affected .
– As e-Passport protocols are dependent on PKI, they may be vulnerable to certificate manipulation and Denial Of Service (DOS) attacks.
– The e-Passport is vulnerable to identity theft since it contains the face shot, the name, and the birthday that help criminal to forge it.

5.3.3 Extended Access Control (EAC)

In 2006, aware about the mentioned weaknesses, the European Union (EU) has issued an e-Passport specification [EU 06], [Kc 05] for Extended Access Control (EAC). To achieve mutual authentication, the EAC proposal introduced two new protocols called Chip Authentication (CA) and Terminal Authentication (TA). These latter are used to improve the capability the Passive Authentication (PA) protocol, the Basic Access Control (BAC) protocol and possibly the Active Authentication (AA) protocol described in the ICAO First generation e-Passport specifications.

The e-Passport's chip can be a contactless smart card containing a Java Card applet (TL ICAO LDS) [Logic 09]. The latter provides the e-Passport services. TL ICAO LDS implements the Basic Access Control (BAC) and the Extended Access Control (EAC) mechanisms which can be both performed based on RSA or ECC algorithm and on the Active Authentication (AA) protocol.

This the order of the four protocols in EAC:
– Basic Access Control (BAC) protocol (mandatory)
– Chip Authentication (CA) protocol (mandatory)
– Passive Authentication (PA) protocol (mandatory)
– Terminal authentication (TA) protocol (optional to access sensitive data)

5.3.3.1 Chip Authentication (CA)

The Chip Authentication (CA) protocol is a *mandatory* protocol in the EAC specifications. It substitutes Active Authentication (AA) protocol as a mechanism to discover forged e-Passports. The Chip Authentication (CA) protocol generates a new pair of encryption and MAC keys to replace BAC's derived ones and enable secure messaging. The keys' generation is done using Diffie-Hellman key agreement protocol. In the ICAO's specification, the e-Passport's chip already has a Chip Authentication (CA) public key (in Data Group 14) and a private key (in secure memory place). When the chip proves that it knows the session key, it is then authenticated by the Inspection System (IS).

5.3.3.2 Terminal Authentication (TA)

The Terminal Authentication (TA) protocol is executed only if the access to secondary biometrical data (such as fingerprint and iris code) is requested. It is a challenge/response mechanism performed by the chip to validate the authenticity of the Inspection System (IS) involved in the Chip Authentication (CA) protocol. Using digital certificates, the IS proves that it is authorized by the home and visiting nations to read e-Passport Chips (more details in Section 5.2.3).

5.3.3.3 Weaknesses

These schemes reinforce the extremely minimal security features offered by the ICAO standards. But, they still have certain shortcomings [Hoepman 06] which are listed in below:

- It is practically impossible to revoke a certificate (There is no time management done by the chip).
- The certificate's hierarchy causes the problem of cross certification among countries.
- There is no information about the way the chip controls the Write Access. The latter is the fact to write an information in the chip and save it, like time of last border crossing...
- Grandmaster Chess Attack has not been addressed.
- Sending identification details during the Chip Authentication (CA) phase leads to Privacy and traceability problems.

In conclusion, the EU approach was designed to allow authentication in offline case for mobile terminals (mobile border inspection units) [Hoepman 06]. In general, the terminals are connected to the network. Therefore, some researchers have proposed an Online terminal authentication.

5.3.4 OSEP Protocol

In 2008, Pasupathinathan et al. [Pasupathinathan 08b] introduced a new solution called On-Line Secure e-Passport Protocol (OSEP). They proposed a mutual authentication between the RFID chip and the Inspection System (IS) [Burmester 09]. Their solution addressed the drawbacks in the current EU's EAC protocol.

The OSEP defined by Pasupathinathan et al. [Pasupathinathan 08b] uses an active monitoring system. The Inspection System (IS) and/or the Document Verifier (DV) could verify if the traveler is in a black list of criminals. They considered the following security features:

- An e-Passport reveals its stored information on the chip only after verification of Inspection System (IS)'s authenticity.
- The OSEP validates the freshness and the authenticity of exchanged messages between participating entities.
- The OSEP preserves ICAO's PKI implementation (first generation e-Passports) and removes cross certification among participating countries as required by EU's EAC (second generation e-Passports).
- It expects the IS to prove public key parameters's correctness to the e-Passport's chip.

Figure 5.1 shows all the entities participating in the solution.

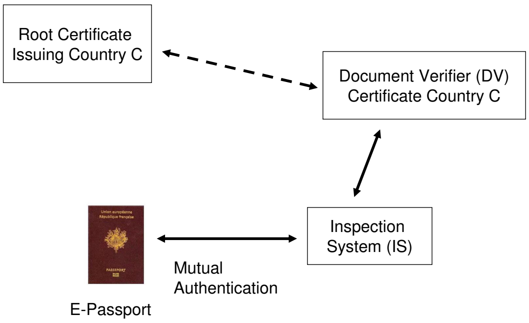


Figure 5.1: Entities Involved in the Mutual Authentication

5.3.4.1 Description of On-Line Secure e-Passport Protocol (OSEP)

The OSEP has 3 phases; *Initial Setup*, *IS Authentication* and *e-Passport Authentication.*

In the Initial Setup (phase 1), all the entities involved in the protocol share the public parameters p, q and g:

- p: 1024-bit prime number,
- q: 160-bit prime number such that $q/(p-1)$,
- g: generator of order q such that $\forall i < q, g^i = 1$ mod p.
- The IS, the DV and the chip C have respectively a public/private key pair: (PK_{IS}, SK_{IS}), (PK_{DV}, SK_{DV}) and (PK_C, SK_C) where for example $PK_{IS} = g^{SK_{IS}}$ mod p (same for the DV and the chip C).

The parameters p, q, g and the public keys PK_{IS}, PK_{DV} and PK_C are signed by the Country Verifying Certificate Authorities (CVCA). They are used by the Diffie-Hellman key Agreement protocol.

In the IS Authentication (phase 2), the chip C and the Inspection System (IS) compute a session key K_{CIS} following these steps.

Step 1: A traveler presents his/her e-Passport to the Inspection System (IS). The IS reads the MRZ information and sends GET CHALLENGE command to the chip C.

Step 2: The chip C generates a secret random number c $(1 <= c <= q-1)$ and calculates $K_C = g^c$. Then, it answers to the GET CHALLENGE command by sending K_C and the public parameters p, q, g to the IS.

Step 3: After receiving the chip C's replay, the IS chooses a random number is $(1 <= is <= q-1)$ and computes $K_{IS} = g^{is}$. The IS creates S_{IS} by signing the message containing MRZ value and K_C.

$$S_{IS} = SIGN\mathrm{SK}_{IS}(MRZ||K_C)$$

The Inspection System (IS) then communicates with the traveler's DV in its proximity and obtains the DV's public key PK_{DV}. The IS encrypts S_{IS}, MRZ information and K_C using PK_{DV}. The message then contains the data encrypted with the IS's certificate signed by CVCA.

Step 4: The DV decrypts the IS's message and verifies $CERT_{CVCA}(PK_{IS}, IS)$ and the signature S_{IS}. If the verification is successful, the DV concludes about the IS's genuineness and produces a new signature S_{DV} to prove IS's authenticity to the chip C.

$$S_{DV} = SIGN\mathrm{SK}_{DV}(MRZ||K_C||PK_{IS})$$

The DV encrypts S_{DV} using public key PK_{IS} and transfers it to the IS. The DV can send in option the chip's public key PK_C.

Step 5: The Inspection System (IS), after decrypting the received message, computes the key $K_{CIS} = K_{IS}^c$. The IS encrypts the signature, MRZ information and K_C using K_{CIS}. It also signs its key K_{IS} and the chip's public parameters using its secret key SK_{IS}. The IS sends its key K_{IS}, the signature and the encrypted value to the chip C.

Step 6: The chip C computes the session key $K_{CIS} = K_C^{is}$. The chip C decrypts the received message using K_{CIS}, retrieve the signature S_{DV} and verifies the signature $SIGN\text{SK}_{IS}(K_{IS}, p, q, g)$. The chip C is convinced of the IS's genuineness after a successful verification. All the next communication are encrypted with the session key K_{CIS}.

Figure 5.2 illustrates the IS Authentication.

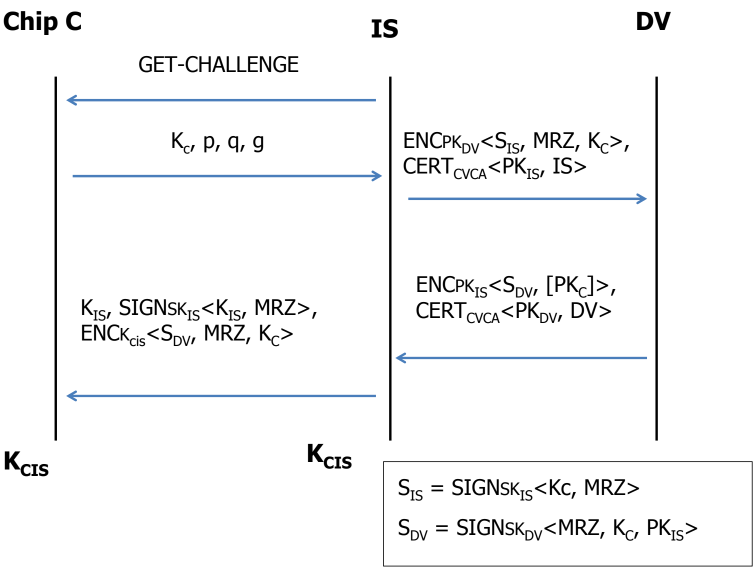


Figure 5.2: IS Authentication

In the e-Passport Authentication (phase 3), the IS verifies:

- the certificate of p, q and g,
- the certificate of the chip's public key PK_C,
- the signature of the MRZ data and K_{CIS}.

After the three phases, the IS is convinced that the e-Passport is genuine and authentic. Figure 5.3 illustrates the e-Passport Authentication.

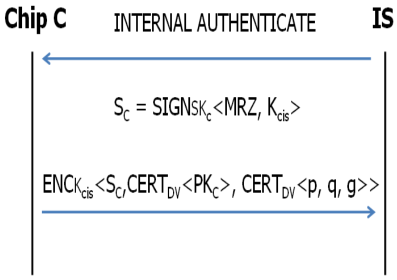

Chip C
INTERNAL AUTHENTICATE
IS
S_C = SIGNSK$_C$<MRZ, K$_{cis}$>
ENCK$_{cis}$<S$_C$,CERT$_{DV}$<PK$_C$>, CERT$_{DV}$<p, q, g>>

Figure 5.3: e-Passport Authentication

5.3.4.2 Security Analysis of the OSEP

We have found some weaknesses in the On-line Secure e-Passport Protocol (OSEP). The first threat the authors wanted to replace is the use of the key defined in Basic Access Control (BAC). these keys hava insufficient entropy as they are derived from the e-Passport number, the date of birth, the date of e-Passport's expiration date and three digits. The International Civil Aviation Organization (ICAO) wanted a personalized key based on information related to the e-Passport's bearer. But, in OSEP, the authors uses Diffie-Hellamn protocol which generates random parameters independent of the traveler's information. We think that if there is no direct relationship between the public parameters used to generate the session key, and the e-Passport bearer's parameters, many e-Passports with the same encryption key can be found. A birthday attack can be done to retrieve these parameters in case of BAC and OSEP.

The second threat happens when OSEP checks only if the e-Passport is genuine and not its bearer. One malicious traveler can use a stolen e-Passport and presents it to the Inspection System (IS). How to make sure the e-Passport's bearer is the genuine bearer as OSEP only checks if the e-Passport is genuine but not its bearer. Moreover, e-Passport parameters do not depends neither on the traveler's data nor on his/her biometric data.

5.4 Our proposed Method using fingerprint

In our solution, we aim at adding the identity verification to the e-Passport's bearer, and we want to use public parameters which are generated using the Biometric Templates (BT). We suppose in our work, that at least one fingerprint is enrolled for each e-Passport holder and its digital value is stored in the Data Group (DG) 3. In [Jain 97], authors insisted on the fact that the quality of acquisition fingerprint images is not good because of distorting and polluted fingerprint.

For each Biometric Template (BT), some minutiae points are extracted from the fingerprint, and we use them to generate an elliptic curve. Assume that each minutiae point is of the form of point *M(x, y)* with abscissa x and ordinate y in [0, 255]. They are coded on 8 bits. Also assume that these points are ordered in the same way in the enrollment and verification phase.

Our solution has three phases:

- The first phase is called the *Initialization phase*. We need to construct an elliptic curve using minutiae's coordinates. This phase is done in a secure way by the e-Passport issuing office which is called the Document Verifier (DV). The latter can be in the native country or an embassy. By doing this phase, we avoid the first threat.
- The second phase is the*Inspection System (IS) Authentication*. We define the session key that is used to secure the communication between the e-Passport's chip and the IS in the visiting country.
- The third phase is the *e-Passport's Bearer Authentication*. The IS checks that e-Passport's bearer is genuine and not a malicious traveler. This eliminates the second threat.

5.4.1 Initialization Phase

In the initialization phase, the e-Passport's issuing office (the Document Verifier (DV)) issues a new e-Passport to the traveler. The latter enrolls his/her fingerprint template which is used to generate elliptic curve domain parameters over $\mathbb{F}_p$. The DV chooses a 256-bit prime number p. Then, it generates an elliptic curve E. The DV chooses also a point $P \in E$ which is the public point for the chip C. The DV stores in the user database the following parameters:

1. ID: identifier;
2. p: 256-bit prime number (modulus);
3. q: 156-bit prime number (order);
4. P: the first public point;
5. A and B: 128-bit numbers, the coefficients of E

The DV also adds the conventional parameters for the e-Passport like name, country, age, gender... Since the MRZ contains 44 characters, we propose to use the 2-D barcodes. They can contains 8192 bytes. This is more than sufficient to store the public cryptographic parameters. These parameters are put in the MRZ:

1. ID: identifier;
2. p: 256-bit prime number(modulus);
3. q: 156-bit prime number (order);
4. P: the second public point.
5. A and B: 128-bit numbers, the coefficients of E

The parameters A, B, p, q and P are certified by the DV. All the procedure is illustrated in the figure 5.4.

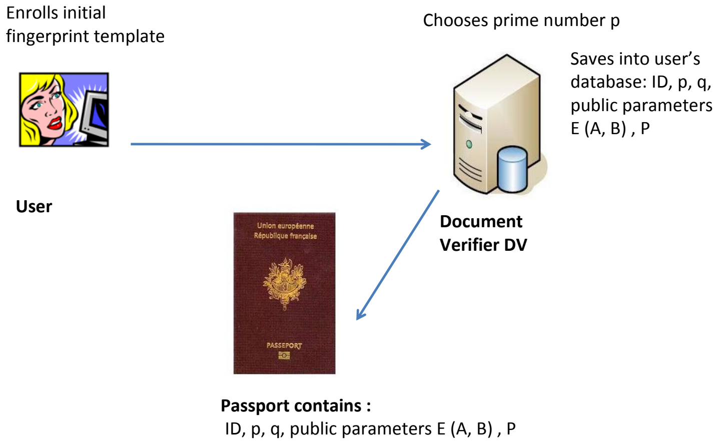

Enrolls initial fingerprint template
User
Chooses prime number p
Saves into user's database: ID, p, q, public parameters E (A, B) , P
Document Verifier DV
Union européenne République française
PASSEPORT
Passport contains :
ID, p, q, public parameters E (A, B) , P

Figure 5.4: Initialization Phase

To choose the elliptic curve $E(y^2 = x^3 + Ax + B)$ defined over $\mathbb{F}_p$, we need to follow the steps in figure 5.5. At the end, an ideal elliptic curve for cryptographic use is obtained. This elliptic curve is used between the chip C and the IS to define session key. After the fingerprint's enrollment, the following steps are done:

In step 1, the Document Verifier (DV) calculates the minutiae points and chooses 32 points of them. For example, if there are 70 minutiae points, it chooses 32 points in ordered manner. Each point $P_i(X_i, Y_i)$ has an abscissa x_i and an ordinate y_i coded on 8 bits. Finally, $P_0(X_0, Y_0)$ is obtained by concatenating all x_i to get $X_0 = x_1||x_2||..........||x_{32}$ and all y_i to get $Y_0 = y_1||y_2||...........||y_{32}$. X_0 et Y_0 are coded on 256 bits.

In step 2, the DV chooses the coefficient $A \in \mathbb{F}_p$. Then, it sets $B = Y_0^2 - X_0^3 - AX_0$, and check that $4A^3 + 27B^2 \neq 0 \text{ mod } p$. (It repeats step 2, if $4A^3 + 27B^2 = 0 \text{ mod } p$).

In step 3, the DV computes $N = Card(E)$ where q is the cardinality (number of the points) of the elliptic curve. If N is not prime, It goes back to step 2; if N is prime, it generates a certificate of primality.

In step 4, the DV checks if $p^j \neq 1 \text{ mod } N$, $\forall 1 <= j <= log_2 p$ (It goes back to step 2, if p has order $<= log_2 p \text{ mod } N$).

After generating the elliptic curve domain parameters over $\mathbb{F}_p$, the DV chooses a point P from the curve E. Then, the e-Passport is ready to be delivered to the traveler.

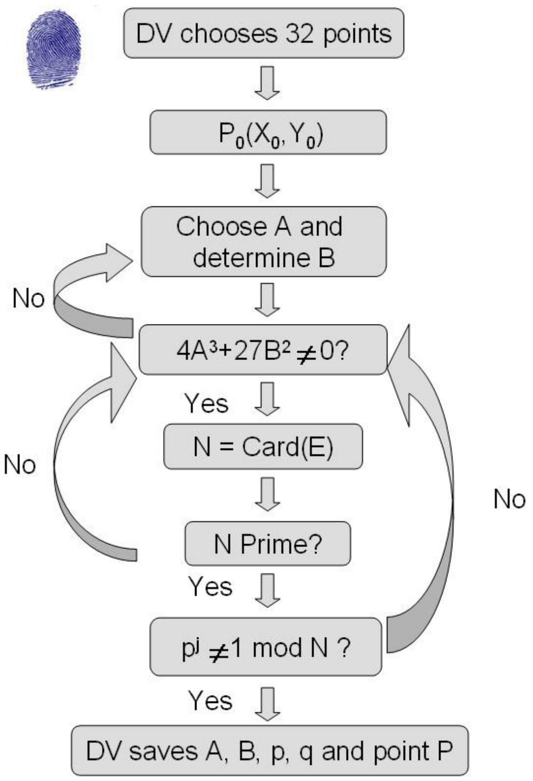


Figure 5.5: Elliptic Key Parameters Generation

The convergence of this loop procedure is limited to two kind of results, either an elliptic curve E is generated or a false error output to declare that no curve (with prime number of point) can be generated. In practice, we can choose a threshold for the number of loop back, so we can change the initial parameters A and B in a reasonable time.

5.4.2 Inspection System (IS) Authentication

IS Authentication is used to verify e-Passport's authenticity. It has the same step as the OSEP' IS authentication defined in Section 5.3.4.1. We propose to change the Diffie-Hellman key agreement protocol by the Elliptic Curve Diffie Hellman (ECDH) Key Agreement defined in the Section 2.6.3.1. The e-Passport's chip C and the IS use the generated elliptic curve parameters to agree on a session key K. Then, the IS starts the e-Passport's bearer authentication. The step of this phase are described in Appendix C.

5.4.3 E-Passport's Bearer Authentication

We want in this phase to verify the identity of the e-Passport's bearer. The latter enrolls his/her fingerprint in the Inspection System (IS) fingerprint sensor. The IS retrieve the 32 minutiae points and calculates the elliptic curve domain parameters using A and p.

First, the IS computes the point $P_0(X_0, Y_0)$ as described in section 5.4.1. It can find a point near to it (minimum Hamming Distance) if the input data is fuzzy. It, then, calculates the coefficient $B = Y_0^2 - X_0^3 - AX_0$. If the IS finds the same B, it means that the e-Passport's bearer is genuine.

At final, the IS and the chip C agree about a session key extracted from K. The chip C can release its data to the IS in a secure way.

5.5 The Proposed Iris Based Authentication Mechanism for e-Passport

This section presents an iris based authentication mechanism for e-Passport. Kanade et al. [Kanade 08] scheme is employed to obtain a key from iris biometrics and this key is used to generate the security parameters. The iris has been described as the best biometric template for biometric comparisons because it has a fine texture that is set randomly during the gestation period. Notwithstanding, monozygotic twins have completely independent iris textures. One of the famous commercially deployed iris recognition algorithm is the John Daugman's IrisCode [Daugman 94].

In the section 5.4, the generation of the elliptic curve domain parameters was based on the enrolled fingerprint data. We present in this section, a modified version based on iris biometrics. We use the stable bit-string extracted from iris code (using the cryptographic key regeneration system of Kanade et al. [Kanade 08]) to generate the elliptic curve parameters. This solution is the result of a collaboration with colleagues from "Electronic and Physic" EPH department of Institut Telecom SudParis [Abid 10]. In the performance phase, we realized the elliptic curve generation and they realized the iris code extraction and key generation. Combining these two work leads to a new elliptic curve generator based on iris code. This work is pioneering in the field as we used intensive tests to verify the algorithms.

5.5.1 Iris Based e-Passport Protocol

The solution has three phases. It begins with the *initialization phase* where an elliptic curve over $\mathbb{F}_p$ is generated, with p being a prime number. The parameters needed to the Elliptic Curve Diffie-Hellman (ECDH) algorithm are saved in the chip. This phase is physically done in the office of the e-Passport's issuing authority.

The second phase is the *Inspection System (IS) authentication*. It is already defined in Section 5.4.2.

The third phase is the *e-Passport's bearer authentication*, where the IS checks that e-Passport's bearer is genuine and not a malicious one.

More details of the three phases are presented below.

5.5.1.1 Initialization Phase

In the initialization phase, the issuing authority (in particular, the Document Verifier (DV)) issues an e-Passport to the traveler. Figure 5.7 presents all the entities participating in this phase.

The traveler who needs a new e-Passport enrolls his/her iris biometrics. To generate the elliptic curve domain parameters, the system takes, as input, the enrolled iris.

The generated elliptic curve E ($y^2 = x^3 + Ax + B \bmod p$) needs to be an ideal elliptic curve for cryptographic use. This elliptic curve is used by the chip and the Inspection System (IS) to define a session key using the Elliptic Curve Diffie-Hellman (ECDH) key agreement protocol.

As shown in Figure 3.12, the security parameters are the key K and the *shuffling key*. The latter is given as input in the verification phase to retrieve the K's value. The *locked* *code* is generated from the key K and the *shuffling key*.

The elliptic curve generation procedure is shown in Figure 5.6.

The key K is hashed with Secure Hash Algorithm (SHA-256) to create the hash value $h_2(K)$. The latter is used with a prime number p coded on 128 bits and a big number A coded on 128 bits to choose the suitable elliptic curve E.

First of all, a $P_0(X_0, Y_0)$ is generated from the hash value $h_2(K)$. Since $h_2(K)$ has 256 bits length, it is cut into two parts, X_0 and Y_0 which are coded on 128 bits.

Then, the DV chooses the coefficient $A \in \mathbb{F}_p$. It sets $B = Y_0^2 - X_0^3 - AX_0$, and check that $4A^3 + 27B^2 \neq 0$. If this condition is satisfied, N = *Card(E)* is computed, where N is the cardinality of the curve. If N is prime, a certificate of primality is generated.

Afterward, the DV checks if $p^j \not\equiv 1 mod N$ for $1 <= j <= log_2 p$. In negative case, the DV starts again the procedure by choosing a new coefficient A.

At the end, the DV obtains an elliptic curve E suitable for cryptographic use. It chooses a point $P \in E$ which is used as the public point of the chip. Then, the e-Passport is ready to be delivered to the traveler.

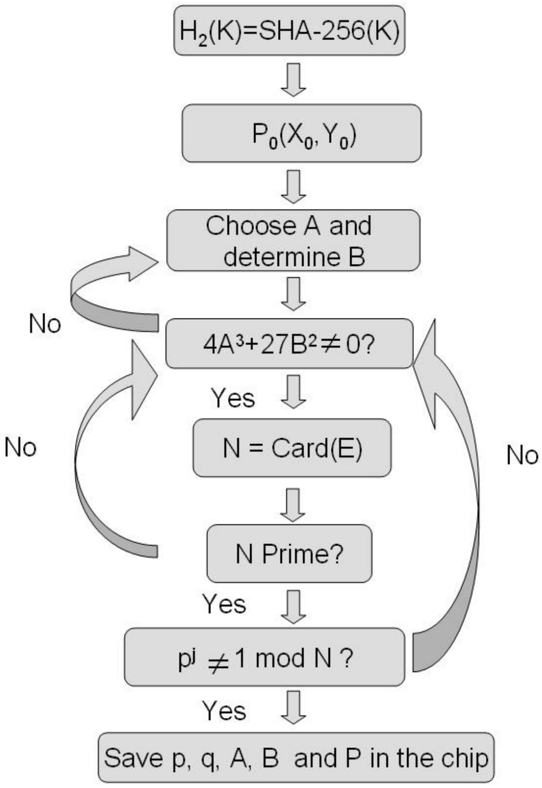


Figure 5.6: Elliptic Curve Generation

The parameters, stored in the user database and the e-Passport chip, are:

- ID: identifier;
- the *locked code*;
- the *shuffling key*;
- $h_1(K)$: the hash value;
- p: 128-bit prime number (modulus);
- q: 80-bit prime number (order);
- the parameters generated using biometric iris:
  - P: the public point;
  - A and B: the coefficients of the elliptic curve E.

The Document Verifier (DV) also adds the conventional parameters for the e-Passport like name, country, age, gender, ...

The parameters A, B, p, q and P are certified by the DV. At the end, the e-Passport is delivered to the traveler. When the validity time of the e-Passport is finished, the bearer can ask for a new one and the system generates a new elliptic curve different from the previous one.

We highlight that the Iris template is not stored in the e-Passport in our proposed solution. The information is stored in protected form (locked code).

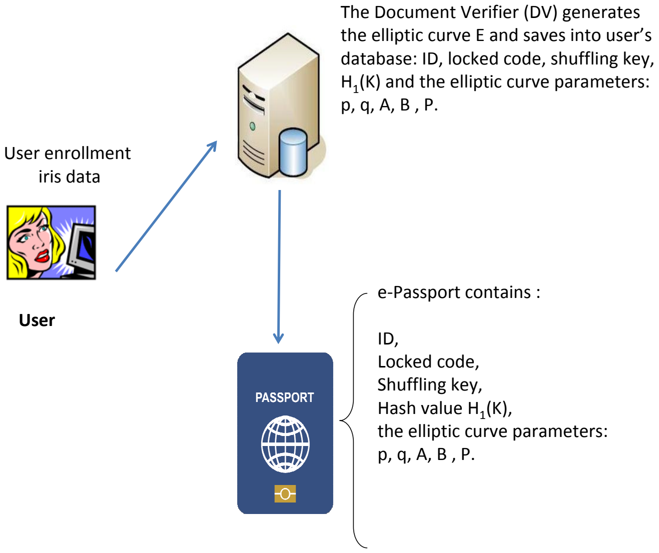

The Document Verifier (DV) generates the elliptic curve E and saves into user's database: ID, locked code, shuffling key, H1(K) and the elliptic curve parameters: p, q, A, B , P.
User enrollment iris data
User
e-Passport contains :
ID,
Locked code,
Shuffling key,
Hash value H1(K),
the elliptic curve parameters:
p, q, A, B , P.
PASSPORT

Figure 5.7: Initialization Phase: Delivering an e-Passport at the Issuing Authority

5.5.1.2 E-Passport's Bearer Authentication

When e-Passport's bearer travels, he/she is asked to prove his/her identity at the border control. The procedure of e-Passport's bearer authentication is showed in Figure 5.8. The traveler provides fresh iris biometrics data. The chip C sends the data needed by the Inspection System (IS) to retrieve the elliptic curve for the authentication of the bearer. These data are, the *locked code*, the *shuffling key*, the hash value $h_1(K)$ and the parameters of the elliptic curve p, A and B.

First, the IS generate *K'* using the fresh iris biometric data, the *shuffling key* and the *locked code*. The IS can check if $h_1(K')$ is equal to $h_1(K)$.

Then, the IS hashes *K'* using SHA-256 to get hash value $h_2(K')$. A point $P'_0(X'_0, Y'_0)$ is created using $h_2(K')$. The point P'_0, p and A are used to generate the elliptic curve *E'* ($y^2 = x^3 + Ax + B' \bmod p$). If the value *B'* is equal to B, the e-passport's bearer is genuine.

In the end, the IS and the chip C agree about a session key extracted from K. The chip C can release its data to the IS in a secure way.

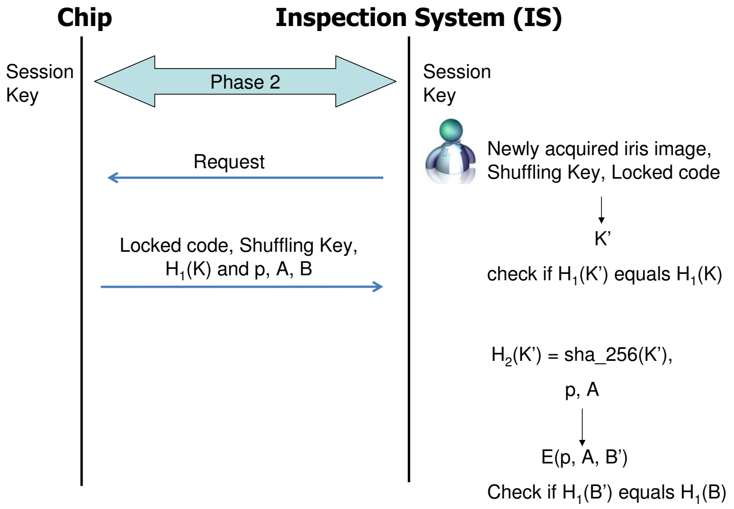

Chip
Inspection System (IS)
Session Key
Phase 2
Session Key
Newly acquired iris image, Shuffling Key, Locked code
Request
K'
Locked code, Shuffling Key, H_1(K) and p, A, B
check if H_1(K') equals H_1(K)
H_2(K') = sha_256(K'),
p, A
E(p, A, B')
Check if H_1(B') equals H_1(B)

Figure 5.8: Procedure of e-Passport's Bearer Authentication at the border control using fresh iris data

5.6 Security Analysis of the Proposed Method

In this section, we analyze the security of our proposed protocol. As our solution is a modification of the Pasupathinathan et al.'s contribution [Pasupathinathan 08b], we solve the same problems. Our improvement depends on the use of the e-Passport's biometrical data in the cryptographic mechanisms. Therefore, we focus in this section on the advantages gained using this method.

From a technical point of view, this solution uses existing ICAO's PKI implementations (first generation e-Passports). The performance of this solution should be better than the On-Line Secure e-Passport Protocol (OSEP) because it uses Elliptic Curve Cryptography (ECC). Thus, the keys are shorter and it is an advantage for the use of RFID Chip which has less memory. The entropy of the keys is better than the one for ICAO. Another benefit is that we avoid storing user database of biometric templates in the Document Verifier (DV)'s server. Since the attack of biometric database is one of the most relevant attacks that we need to prevent.

Our solution strengthen the Data Confidentiality since the Elliptic Curve Diffie-Hellman (ECDH) Key Agreement is more secure that the conventional Diffie-Hellman protocol. It is difficult to retrieve the session key because of the discrete logarithm criterion of Elliptic curve.

To prevent Grandmaster Chess Attack, we presented the e-Passport's Bearer Authentication. When an attacker uses a Biometric Template (BT) different from the one stored in the e-Passport's Data Structure, the IS defines a curve $E' \neq E$ and rejects the traveler. To ensure a mutual authentication, the e-passport's chip C relay the authentication of the IS to the DV by checking the certificates signed by the CVCA.

As an e-Passport has an expiration date, a new elliptic curve and public parameters can be generated. This leads to a fresher key and avoids replay attacks after the expiration date. Let X_0 and Y_0, coded on 256 bits, be the coordinates coded on 256 bits of point Q. The session key, generated from Q is longer than the session key generated after the Basic Access Control (BAC) which is 56-bit long with an entropy equal to 52 bits.

The e-Passport's Bearer Authentication is good in case of False Acceptance but in the case of False Rejection, the presence of human assistance to the IS is a plus. The fuzziness of BT can lead to false rejection of genuine travelers. Working with face can lead to the same problems if the traveler changes his/her look by having a beard, a long hair or glasses.

5.7 Implementation and Performance Evaluation

This section focuses on the evaluation of phase 1 and 3 especially to compute the False Rejection Rate (FRR) and the False Acceptance Rate (FAR).

We implemented the process to generate the elliptic curve using the iris code (phase 1) and the process to verify the identity of the e-passport's bearer (phase 3).

5.7.1 Implementation of the Elliptic Curve Generator

In order to evaluate the performance of our solution, we implemented the cryptographic algorithms using C/C++ language including *Multiprecision Integer and Rational Arithmetic C/C++ Library* (MIRACL) [MIRACL]. This library offers some cryptographic routines like SHA-256 and Advanced Encryption Standard (AES). It offers also elliptic curve routines. We used the Schoof-Elkies-Atkin (SEA) Algorithm [Schoof 95] already implemented in MIRACL. It is used to count points on $\mathbb{F}_p$ for elliptic curve E ($y^2 = x^3 + Ax + B \bmod p$). A prerequisite for the Elliptic Curve Cryptography (ECC) is that the order of the elliptic curve should be prime.

The iris based key regeneration scheme was implemented in MATLAB. The code uses an iris code as input, a key K and the *shuffling key*. The output is the *locked code*.

With an Intel Xeon CPU E5430 @ 2,66 GHz, it takes at most 3 minutes to generate the elliptic curve in phase 1 and about 6 to 7 seconds to retrieve the curve from the e-Passport's Bearer Authentication (phase 3).

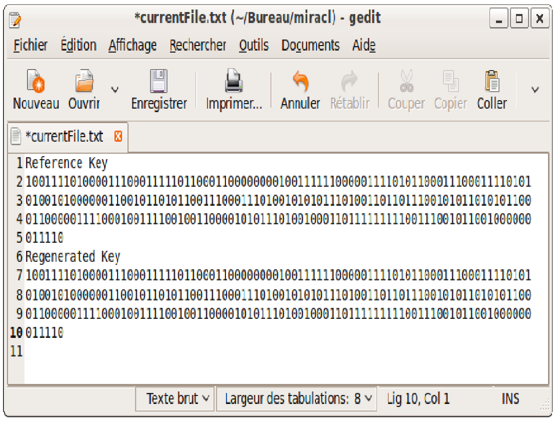
5.7.2 Performance Evaluation

There is a free Implementation of Machine Readable Travel Documents [JMRTD 06], but we are not interested in evaluating the protocol itself. Instead, we want to have a biometric evaluation of the solution. This means: is it feasible to use the biometric data for such a cryptographic protocol and with the smaller possible FRR and FAR ?

The performance of the system is evaluated using the publicly available *National Institute of Standards and Technology-Iris Challenge Evaluation (NIST-ICE)* database [NIST]. This database contains 2,953 images from 244 different eyes. The experimental protocol given with the NIST-ICE's database [NIST] has two different experiments for right and left eye images. This protocol results in more than one million comparisons for each of these experiments. Considering the high amount of time required for each comparison with the proposed system, it is not feasible to carry out all these comparisons.

Hence, a subset of the database was selected such that there are strictly five images per eye. Using this criterion, we created the dataset with 875 iris images coming from 175 different eyes.

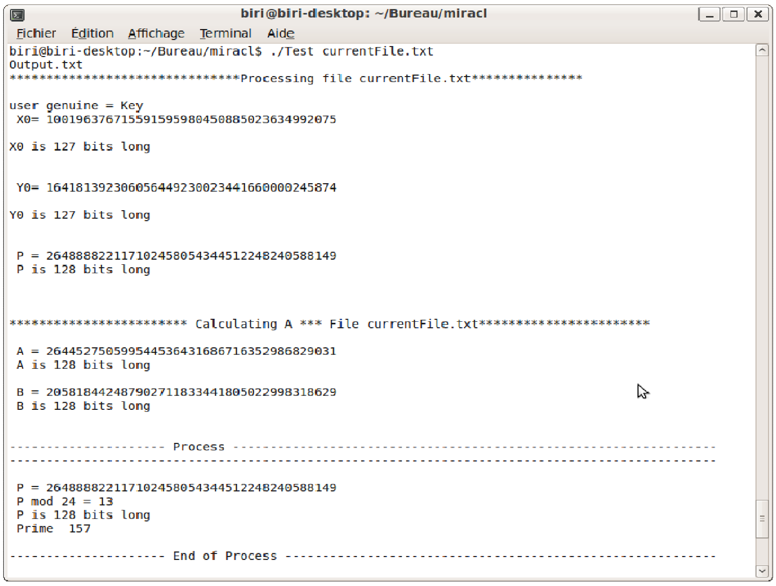
In order to carry out genuine user comparisons, we compared each image of an eye with every other images of that eye. This results in ten genuine comparisons per eye, totaling 1,750 genuine comparisons. We show an example of genuine test. Figure 5.9 presents the reference key and the regenerated key. The latter is used to generate the elliptic curve parameters of a genuine user with $t_s = 10$.



```
Reference Key
10011110100001110001111101100011000000001001111110000011110101100011100011110101
01001010000001100101101011001110001110100101010111010011011011100101011010101100
01100000111100010011111001001100001010111010010001101111111110011100101100100000
011110
Regenerated Key
10011110100001110001111101100011000000001001111110000011110101100011100011110101
01001010000001100101101011001110001110100101010111010011011011100101011010101100
01100000111100010011111001001100001010111010010001101111111110011100101100100000
011110
```

Figure 5.9: Reference and Regenerated Keys

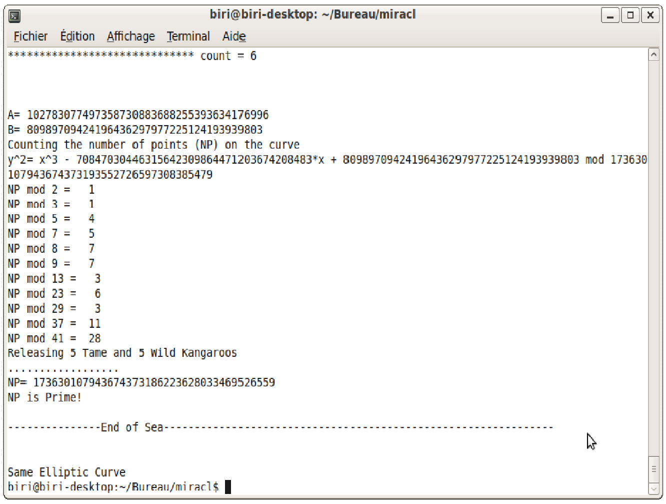
Figure 5.10 presents the phase where X_0 and Y_0 are computed. In the enrollment phase, the system chooses a prime number p and a coefficient A and then, calculates $B = Y_0^2 - X_0^3 - AX_0$. In the verification phase (phase 3), p and A are the input of the program and only B is computed. Also, image 2 of the same genuine user is chosen as input.


biri@biri-desktop: ~/Bureau/miracl
Fichier Édition Affichage Terminal Aide

```
biri@biri-desktop:~/Bureau/miracl$ ./Test currentFile.txt
Output.txt
*******************************Processing file currentFile.txt***************

user genuine = Key
 X0= 100196376715591595980450885023634992075

X0 is 127 bits long


 Y0= 164181392306056449230023441660000245874

Y0 is 127 bits long


 P = 264888822117102458054344512248240588149
 P is 128 bits long



************************ Calculating A *** File currentFile.txt***********************

 A = 264452750599544536431686716352986829031
 A is 128 bits long

 B = 205818442487902711833441805022998318629
 B is 128 bits long


--------------------- Process ---------------------------------------------------------------
-------------------------------------------------------------------------------------------------


 P = 264888822117102458054344512248240588149
 P mod 24 = 13
 P is 128 bits long
 Prime  157

--------------------- End of Process ----------------------------------------------------------
```

Figure 5.10: Generation of Elliptic Curve Parameters

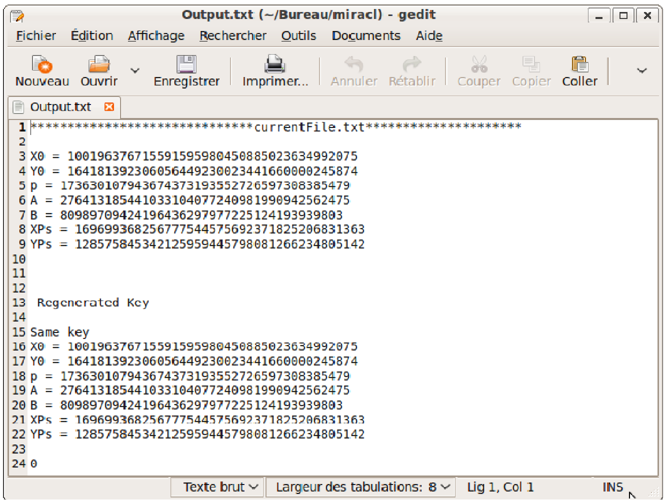
Figure 5.11 presents the end of the elliptic curve generator program. In this case, the same curve is retrieved.


biri@biri-desktop: ~/Bureau/miracl
Fichier Édition Affichage Terminal Aide

```
****************************** count = 6




A= 102783077497358730883688255393634176996
B= 80989709424196436297977225124193939803
Counting the number of points (NP) on the curve
y^2= x^3 - 70847030446315642309864471203674208483*x + 80989709424196436297977225124193939803 mod 173630
107943674373193552726597308385479
NP mod 2 =   1
NP mod 3 =   1
NP mod 5 =   4
NP mod 7 =   5
NP mod 8 =   7
NP mod 9 =   7
NP mod 13 =   3
NP mod 23 =   6
NP mod 29 =   3
NP mod 37 =  11
NP mod 41 =  28
Releasing 5 Tame and 5 Wild Kangaroos
..................
NP= 173630107943674373186223628033469526559
NP is Prime!

---------------End of Sea---------------------------------------------------------------


Same Elliptic Curve
biri@biri-desktop:~/Bureau/miracl$ 
```

Figure 5.11: Same Elliptic Curve Regenerated

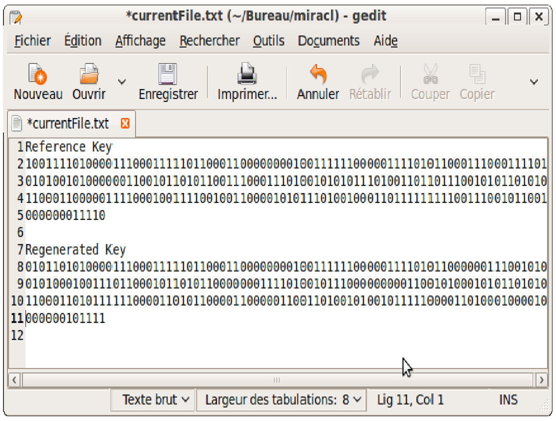
Figure 5.12 presents the output file where there are the parameters of the reference curve and the regenerated one. In this case, the same curve is obtained in the verification phase.



```
******************************currentFile.txt********************

X0 = 100196376715591595980450885023634992075
Y0 = 164181392306056449230023441660000245874
p = 173630107943674373193552726597308385479
A = 276413185441033104077240981990942562475
B = 80989709424196436297977225124193939803
XPs = 169699368256777544575692371825206831363
YPs = 128575845342125959445798081266234805142



 Regenerated Key

Same key
X0 = 100196376715591595980450885023634992075
Y0 = 164181392306056449230023441660000245874
p = 173630107943674373193552726597308385479
A = 276413185441033104077240981990942562475
B = 80989709424196436297977225124193939803
XPs = 169699368256777544575692371825206831363
YPs = 128575845342125959445798081266234805142

0
```

Figure 5.12: The Output File in Genuine Verification

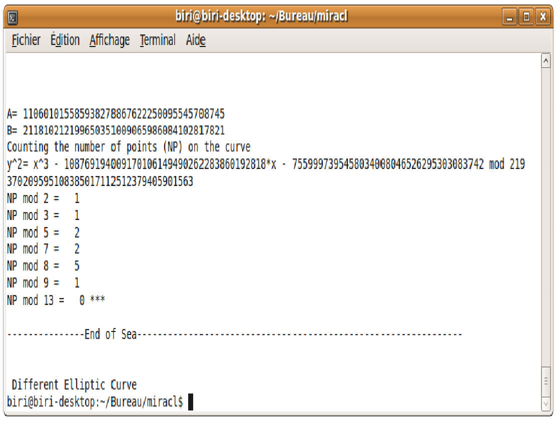
For impostor comparisons, we carried out every possible combination of image pairs coming from two different eyes. This comes out to be 308,625 comparisons. Figure 5.13 presents the reference key and the regenerated key. The latter is used to generate the elliptic curve parameters of an impostor user with $t_s = 10$.



```
Reference Key
1001111010000111000111110110001100000000100111111000001111010110001110001111101
0101001010000000110010110101100111000111010010101011101001101101110010101101010
110001100000111100010011110010011000010101110100100011011111111100111001011001
000000011110

Regenerated Key
010110101000011100011111011000110000000010011111100000111101011000000111001010
010100010011101100010110101100000001111010010111000000000110010100010101101010
110001101011111100001101011000011000001100110100101001011111000011010001000010
000000101111

```

Figure 5.13: Reference and Regenerated Keys for impostor case

Figure 5.14 presents the end of the elliptic curve generator program. In this case, a different curve is retrieved.

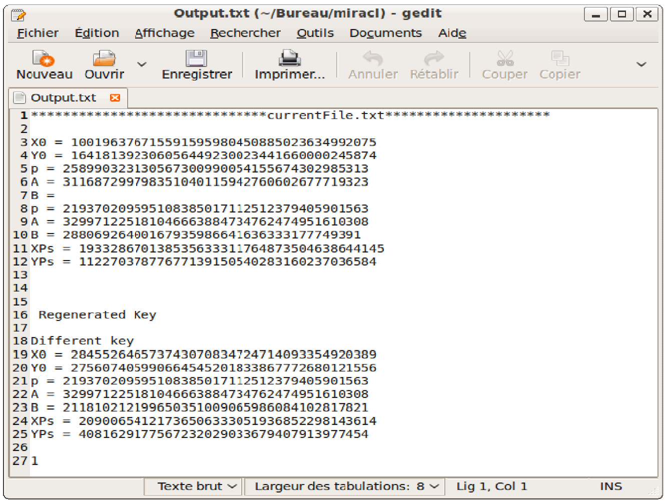


biri@biri-desktop: ~/Bureau/miracl
Fichier Édition Affichage Terminal Aide

```
A= 110601015585938278867622250095545708745
B= 211810212199650351009065986084102817821
Counting the number of points (NP) on the curve
y^2= x^3 - 108769194009170106149490262283860192818*x - 75599973954580340080465262953O3083742 mod 219
370209595108385017112512379405901563
NP mod 2 =   1
NP mod 3 =   1
NP mod 5 =   2
NP mod 7 =   2
NP mod 8 =   5
NP mod 9 =   1
NP mod 13 =   0 ***

--------------End of Sea------------------------------------------------------------


 Different Elliptic Curve
biri@biri-desktop:~/Bureau/miracl$
```

Figure 5.14: Different Elliptic Curve Regenerated

Figure 5.15 presents the output file where there are the parameters of the reference curve and the regenerated one. In this case, a different elliptic curve is obtained in the verification phase. There is also a remark concerning the computation of B, where the first choice of p and A was not successful. This leaded to new choice of these values and hence a new B is obtained.



Output.txt (~/Bureau/miracl) - gedit
Fichier Édition Affichage Rechercher Outils Documents Aide
Nouveau
Ouvrir
Enregistrer
Imprimer...
Annuler
Rétablir
Couper
Copier
Output.txt
Texte brut
Largeur des tabulations: 8
Lig 1, Col 1
INS

```
****************************currentFile.txt********************

X0 = 100196376715591595980450885023634992075
Y0 = 164181392306056449230023441660000245874
p = 258990323130567300990054155674302985313
A = 311687299798351040115942760602677719323
B =
p = 219370209595108385017112512379405901563
A = 329971225181046663884734762474951610308
B = 288069264001679359866416363331777 49391
XPs = 193328670138535633311764873504638644145
YPs = 112270378776771391505402831602370365 84



 Regenerated Key

Different key
X0 = 284552646573743070834724714093354920389
Y0 = 275607405990664545201833867772680121556
p = 219370209595108385017112512379405901563
A = 329971225181046663884734762474951610308
B = 211810212199650351009065986084102817821
XPs = 209006541217365063330519368522981436 14
YPs = 408162917756723202903367940791397745 4

1
```

Figure 5.15: The Output File in Impostor Verification

We carried out the experiments for two setting of the *error correction capacity* t_s of Hadamard code (more details are given in [Kanade 08]). The length of the key K for t_s

= 10 is 247 bits while for t_s = 15, it is 186 bits. The results for Key regeneration system of Kanade et al. and Iris based e-Passport authentication scheme are shown in Table 5.2.

t_s	Key regeneration system of Kanade et al.	Iris based e-Passport authentication scheme
10	FAR=0.0005% and FRR=2.46%	FAR=0.01% and FRR=5.26%
15	FAR=0.21% and FRR=0.86%	FAR=0.20% and FRR=3.60%

Table 5.2: Results on NIST-ICE database : subset of the database ; 175 eyes, 5 images each

It can be seen from Table 5.2 that the value of FAR rises with the increase in the error correction capacity t_s. On the other hand, with higher values of t_s, low FRR can be obtained. This is consistent with the fact that t_s acts as a threshold for the biometric data variability. The more the value of t_s, the more errors are allowed in the biometrical data. Thus, when high security is required, lower values of t_s shall be selected.

In our case, we make experiments in order to study the feasibility of our solution and to compare our work with other biometric systems from a performance point of view. If we compare our results with the result of the Key regeneration system of Kanade et al., we see that their work is better because it doesn't include the elliptic curve generator which add more error rate in the system. But, Our performance values are in the average range of the experimental results presented in Section 3.5. Thus, we consider our results as satisfying for the e-Passport scenario.

5.8 Conclusion

Many solutions were proposed to secure the e-Passport protocols. The International Civil Aviation Organization (ICAO) provided the standard security measures which are implemented in many countries like USA and Australia. The first purpose was to enhance security at border controls and, secondly, to perform these security checks in automatic manner. However, many parts of the standard may be subject to attacks that aim at recovering information from the e-Passport's bearer. The European Union (EU) provided an enhancement to the e-Passport protocol by defining Extended Access Control (EAC). This scheme has an important problem related to cross certification among countries. In 2008, Pasupathinathan et al. [Pasupathinathan 08b] designed a new solution called On-Line Secure e-Passport Protocol (OSEP). They proposed a mutual authentication between the e-Passport's chip and the Inspection System (IS). Their solution addressed the drawbacks in the EU's EAC protocol.

Our new method was based on the OSEP protocol and we used the Elliptic curve Diffie-Hellman (ECDH) Key Agreement protocol to define the session key. We used the fingerprint and the iris code to generate the elliptic curve domain parameters.

We analyzed the security of our solution. We found that our solution fulfills its goals and prevents the system from the attacks. The use of biometric in the cryptographic solution is, in our point of view, a very important issue as this biometrical data is stored in the e-passport's chip without a direct link to the security.

This solution is validated by using iris biometrics. We performed tests on the National Institute of Standards and Technology-Iris Challenge Evaluation (NIST-ICE) database of iris images. We computed the False Rejection Rate (FRR) and the False Acceptance Rate (FAR). The results obtained (e.g., FRR of 0.2% and FAR of 3.6%) are satisfying and the use of iris biometrics is encouraging for the deploying of this solution.

In the end of this chapter, we remind that the combination of cryptography and biometry is a prominent research field and many works have to be done in the near future. In the next chapter, we consider a new type of identity which is any simple string like e-mail address. We focus on using this identity to enhance the authentication mechanisms in the Next generation network (NGN).

Chapter 6

A new service Authentication Mechanisms for IP Multimedia Subsystem (IMS)

6.1 Introduction

The authentication is the process that helps to decide if a user is who he/she claims to be. Authentication is achieved using something the user knows (e.g. password), something the user has (e.g. security token) or something the user is (e.g. biometrical data).

The authentication process prevents many threats especially identity fraud. The systems with high threat level require different forms of authentication to confirm the users' digital identity. For systems with low threat level, the confirmation of the digital identity is not as important from a risk point of view.

Authentication mechanisms rely on the identity verification and the registration processes. The identity registration process usually involves a user with the operator or company authentication mechanisms. The identity can be an identifier ID and a password, a security token, a digital certificate and/or some of the user's biometrical data.

In the chapters 4 and 5, we focused on the use of the biometrics in the authentication mechanisms. In this chapter, we are interested in the Next Generation Network (NGN) framework. The IP Multimedia Subsystem (IMS) [Tirado 08] is a standardized NGN architecture defined by the European Telecommunications Standards Institute (ETSI) and the 3rd Generation Partnership Project (3GPP) [Camarillo 04].

The IMS is an overlay architecture which is built on top of the IP network. It is designed to provide multimedia services (such as Voice over IP (VoIP), video conferencing, press-to-transmit, etc...) on top of all IP networks as well as Next Generation Networks (NGN). We can read in [Poikselka 09] that "The IMS is thus promising in the future technology for the convergence of data, speech and mobile networks, thanks to providing easy and efficient ways to integrate different value added services and seamless integration of legacy services".

In the IMS, the user's identity is usually used in the authentication procedure as an identifier (login) and not a part of the password or the cryptographic key. Each user or subscriber uses an IMS-Subscriber Identity Module (ISIM) card with a stored secret key to be authenticated to the IMS network and to be able to access the IMS services.

Indeed, the IMS's authentication proved to have some security limitations such as server spoofing and off-line password guessing attacks [Niemi 02]. Another requirement provided in the IMS, is the tight attachment of the user's authentication to the user equipment (UE) since the ISIM card is used as a substrate (the element communicating with the network) and also authentication is restricted to one algorithm: Authentication and key Agreement (AKA) [Tirado 08]. This fact limits the personalized access and hence the service personalization, since each user should have access through his/her own devices (having his/her own ISIM card) in order to be correctly identified. As a consequence, IMS authentication don't realize authentication in a personalized manner, which is an important prerequisite in new services such as social Internet ones. The strong dependency on the ISIM card also limits the security performance [Priselac 08] and stands as an obstacle towards compatibility and evolution (considering that many operators do not support the smart card).

On the other hand, using the Authentication and key Agreement (AKA) protocol in IMS proved to have some weakness, like short key for cryptographic purposes [Priselac 08], [S.R0086-B 05]. Many solutions are proposed to strengthen IMS's security like in [Wu 09] where the authors define a new AKA based on Elliptic Curve Cryptography (ECC) and in [Huang 07] where the authors define a new AKA called one pass AKA for the Universal Mobile Telecommunications System (UMTS). Furthermore, Ring et al. [Ring 06] tried to design a new AKA mechanism for Session Initiation Protocol (SIP) using Identity Based Cryptography (IBC) [Boneh 01]. However, these works focus on the subscriber's authentication with nothing special on service authentication. In this chapter, we want to modify the IMS's service authentication mechanism by involving the Identity Based Cryptography (IBC), presented in Section 2.7. The goal is to personalize the service authentication mechanism.

The remainder of this chapter is organized as follows, Section 6.2, describes the IMS authentication process. In Section 6.3, we present our novel solution, and in Sections 6.4 and 6.5, we present a security analysis and performance evaluation respectively for our proposed solution. Finally, we conclude the chapter in Section 6.7 and highlight some points for future work.

6.2 Overview on IP Multimedia Subsystem (IMS)

The IP Multimedia Subsystem (IMS) is an overlay architecture which is built on top of all IP networks as well as Next Generation Networks (NGN). It enables various types of multimedia services to end-users using common Internet-based protocols [Camarillo 04]. It was originally designed by the wireless standards body 3^{rd} Generation Partnership Project (3GPP) [3GPP] and was later extended by the Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN) as a subsystem of NGN [TISPAN]. The IMS supports IP Multimedia applications such as video, audio and multimedia conferences. The Session Initiation Protocol (SIP) was chosen as the signaling protocol for starting and ending multimedia sessions. The security of IMS services, authentication, authorization protocols and encryption/decryption procedures, have been defined and implemented [33.102 09], [33.203 09], [33.210 09].

Figure 6.1 illustrates the main entities constituting the IMS core.

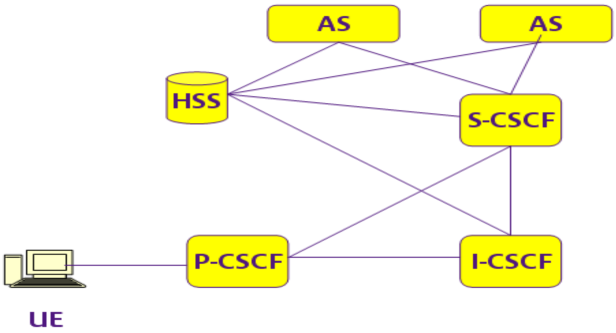


Figure 6.1: IMS Architecture

Since IMS uses Session Initiation Protocol (SIP) for the control and signaling of sessions, its main architectural elements are SIP proxies, known as Call Service Control Functions (CSCF). The CSCFs handle all the SIP session signaling and are divided into P(proxies)-CSCF, I(interrogating)-CSCF and S(serving)-CSCF.

1. the *P-CSCF* is used as IMS contact points for end users within IMS.
2. the *I-CSCF* is the contact point within the operator's network and forwards connections to the appropriate destination.
3. the *S-CSCF* is considered as the focal entity of the IMS since it is responsible for users' authentication, registration and authorization, and also for managing the application servers (AS).

The Home Subscriber System (HSS) is another important entity in IMS which is a database for all subscribers and service-related data of the IMS. The main data stored in the HSS includes user identities, registration information, etc.

The next sections present respectively the Subscriber Identification, the Subscriber Authentication and the Service Authentication in IMS.

6.2.1 Subscriber Identification in IMS

In IMS, each user/subscriber has two types of identities:

1. An IP Multimedia Private Identity (IMPI), where every IMS's user shall have one or more Private User Identities, but there is only one private user identity stored in each ISIM card. The home network operator assigns the private identity for being used in registration, authentication, authorization, administration, and accounting purposes. The private user identity has the form of a Network Access Identifier (NAI).
2. An IP Multimedia Public Identity (IMPU), where every IMS's user shall have one or more Public User Identities and each ISIM card stores at least one public user identity. The Public User Identities are used by any user for requesting communications to other users and to services access. The Public User Identity takes the form of a Session Initiation Protocol (SIP) Uniform Resource Identifier (URI) or the "tel:"-URI format.

Figure 6.2 [33.228 09] shows the relationship between the IMPI, the IMPU and the services.

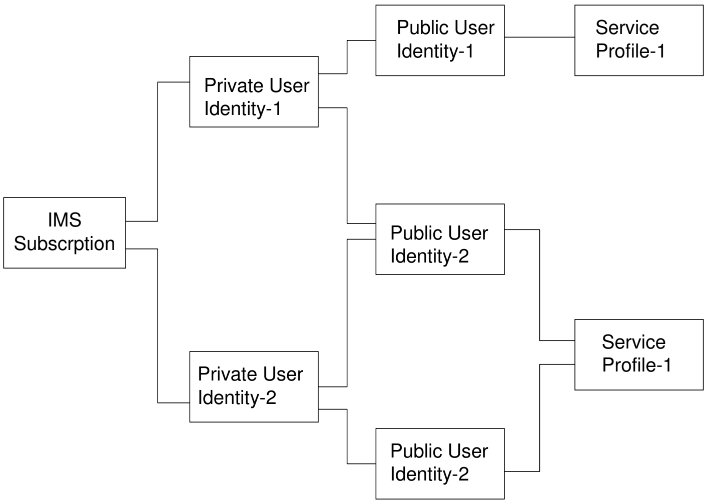


Figure 6.2: Relationship between IMS Identities

In this figure, the Service Profile is a collection of services and the user's data are stored in the HSS. Each IMPU is associated with one and only one service profile; however each service profile can be associated with one or more IMPU. Different IMPIs can share the same IMPU within the same IMS subscription. Hence, an IMPU can be simultaneously registered from multiple UEs that use different IMPIs and different contact addresses.

6.2.2 Subscriber Authentication in IMS

Authentication and key Agreement in the IMS is called IMS AKA, which is based on a secret shared key *sk*, shared between the user (ISIM card) and the Home Subscriber Server (HSS) in the Homework. The secret key (K) and AKA's parameters are stored in IP Multimedia Services Identity Module (ISIM) which is normally embedded on the Universal Integrated Circuit Card (UICC) like a smart card based device. This authentication in IMS is directly coupled to the SIP registration process. As the HSS can not communicate directly with the User Equipment (UE), it is the S-CSCF that performs the authentication process.

Figure 6.3 shows the subscriber's authentication within the IMS core network.

In order to be authenticated, the UE sends the IMPU stored in the ISIM card in the initial REGISTER request (message 1).

Upon the reception of the REGISTER request, the P-CSCF examines the “home domain name” to discover the I-CSCF (message 2).

Then, this latter sends an information query to the HSS to find the appropriate S-CSCF, and then forwards the REGISTER request to the S-CSCF (messages 3-5).

When receiving this REGISTER request, the S-CSCF downloads an array of n (n>=1) Authentication Vectors (AVs) from the HSS (messages 6-7). The AVs are ordered based on sequence numbers (SQN) and each AV includes a random challenge (RAND), the expected result (XRES), the network authentication token (AUTN), the Integrity Key (IK) and the Ciphering Key (CK).

In order to authenticate the UE, the S-CSCF sends an authentication request within a “401 (Unauthorized) response”, which includes: the RAND, the AUTN, the IK and the CK (messages 8-9).

The P-CSCF, when receiving the “401 Unauthorized Response”, removes the IK and the CK from the response before relaying it to the UE (message 10).

After receiving the response, the UE first retrieves the SQN from the received RAND and computes the eXpected Message Authentication Code (XMAC). Then, the UE compares the XMAC with the Message Authentication Code (MAC) which is included in AUTN. If they are identical and if the SQN is in correct range, the network is authenticated by the UE. Then the UE calculates the challenge responses RES and sends it to the S-CSCF (messages 11-13). The UE also calculates the resulting IK which is then shared between the P-CSCF and the UE. When the S-CSCF receives the response, it compares the RES and XRES that were received in the AV from the HSS. The UE’s authentication process ends up as successful with either a “SIP OK” message or a “SIP unauthorized message”, both messages are sent by the S-CSCF (messages 14-16).

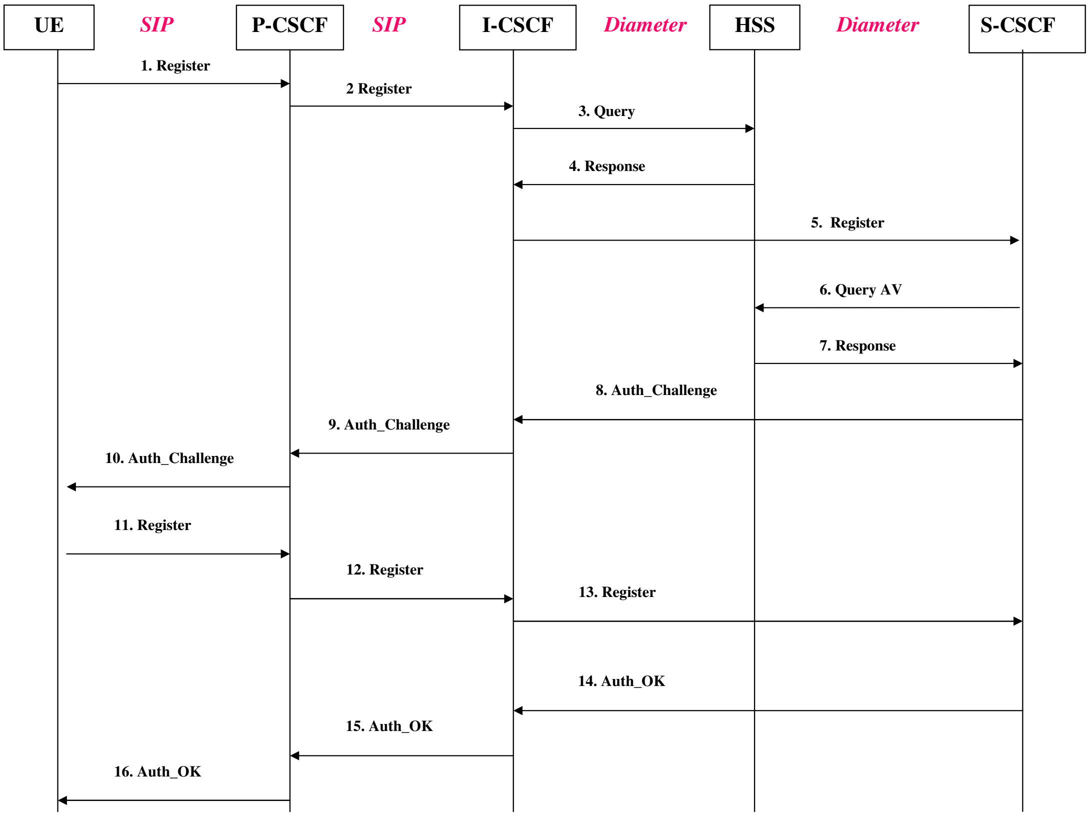


Figure 6.3: Subscriber Authentication within the IMS Core Network

6.2.3 Service Authentication in IMS

Parallel to SIP traffic, numerous services might be accessed over the Hypertext Transfer Protocol (HTTP). In order to allow the access to services over HTTP in a secure manner, the IMS uses the Generic Bootstrapping Architecture (GBA) [33.220 09], [Sher 06] and [Sher 07]. There are four entities participating in the process:

- the User Equipment (UE)
- the Network Application Function (NAF)
- the Bootstrapping Server Function (BSF)
- the Home Subscriber Server (HSS)

Figure 6.4 shows the network model of the entities involved in the bootstrapping approach, and the reference points used between the entities.

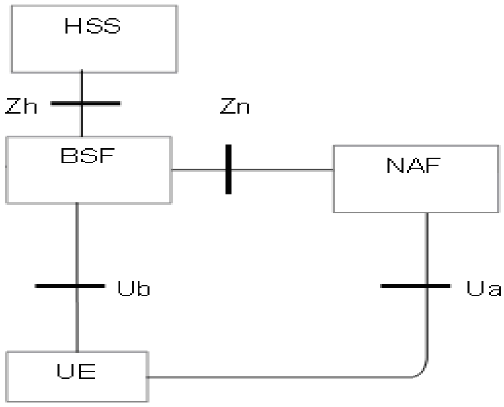


Figure 6.4: Generic Bootstrapping Architecture (GBA)

6.2.3.1 Network Elements

The GBA consists of four entities:

6.2.3.1.1 User Equipment (UE)

An UE is a Universal Integrated Circuit Card (UICC) containing an Universal Subscriber Identity Module (USIM) or an IMS-Subscriber Identity Module (ISIM) related information. The UE shall:

- execute the HTTP Digest AKA protocol;
- use the USIM and the ISIM in bootstrapping;
- choose the USIM or the ISIM when both of them are present;
- derive a new key material to be used over Ua interface from the shared keys IK and CK;
- support the NAF-specific application protocols.

6.2.3.1.2 Network Application Function (NAF)

The NAF needs to fulfill these requirements:

- the UE and the NAF do not have a previous association before the GBA starting;
- the NAF shall determine the BSF's location and communicate securely with it;
- the NAF shall obtains the shared key materiel defined between the UE and the the BSF during the protocol exchanges;

– the NAF shall get the User Security Settings (USS) from the HSS via the BSF.

6.2.3.1.3 Bootstrapping Server Function (BSF)

The generic Bootstrapping Server Function (BSF) is a central entity in the GBA. Its functionalities consist of:

- the mutual authentication between it and the UE using the AKA protocol;
- the agreement on session keys that are afterwards applied between the UE and the NAF;
- the restriction of the key material's use only to a specific NAF by performing the key derivation procedure. The latter may be used with multiple NAFs in the same session
- the acquirement of the GBA user security settings (GUSS) from the HSS;
- the maintaining of a NAFs' list which is used to make correspondence between the GUSS and the NAF capabilities.

6.2.3.1.4 Home Subscriber System (HSS)

The set of all GBA user security settings (GUSS), is stored in the HSS. The subscriber can have multiple subscriptions. In this case, the HSS may contain one or more GUSSs that can be mapped to one or more private identities (IMPI). The latter shall only have zero or one GUSS mapped to it. The Authentication, Authorization and Accounting (AAA) protcol(already presented in Section 3.5.4) in the IMS is the DIAMATER protocol [Calhoun 03] and it is implemented in the Home Subscriber Server (HSS).

The GUSS shall be composed by some parameters useful for the BSF like:

- the type of the UICC in the UE;
- the subscriber's key lifetime;
- optionally, the timestamp to mark when the GUSS has been last modified by the HSS.

6.2.3.2 Reference Points

We describe in this paragraph the four reference points in the system.

6.2.3.2.1 Reference point Ub

The reference point Ub offers mutual authentication between the UE and the BSF. The BSF sends a bootstrapping transaction identifier (B-TID) to the UE. The AKA protocol is used on the reference point Ub to establish shared keys.

6.2.3.2.2 Reference point Ua

The application protocol,which is secured using the keys material agreed between the UE and the BSF, is carried by the reference point Ua. This key material was generated using the HTTP Digest AKA over reference point Ub. The UE and the NAF shall be able to secure the reference point Ua using the GBA-based shared secret;

6.2.3.2.3 Reference point Zh

The reference point Zh is between the BSF and the HSS. The Zh security requirements are the mutual authentication, the confidentiality and the integrity. Theses requirements may be fulfilled by security measures chosen by the Network operator since the BSF and the HSS are its entities. The DIAMETER Protocol is chosen to secure the management of the clients on the HSS's database and by the way, secure the communication on the reference point Zh. The reference point Zh is needed to carry the Authentication Vector (AV) and the GBA User Security Setting (GUSS) from the HSS to the BSF.

6.2.3.2.4 Reference point Zn

The reference point Zn is between the BSF and the NAF. The latter requests the key material from the BSF. The request contains NAF's public hostname used by the UE's corresponding request. The BSF verifies the NAF's authorization to use this hostname, (checks the Fully Qualified Domain Name (FQDN)provided by the UE). The, the BSF sends the key material and the GUSS over the reference point Zn to the NAF.

6.2.3.3 The GBA Authentication Protocol

The Generic Bootstrapping Architecture (GBA) performs authentication between the BSF and the UE, which is also based on AKA. Figure 6.5 shows the GBA authentication for services access in IMS.

In message 1, to grant service access to UE, the latter communicates over Ua with the NAF whithout sending GBA's parameters.

In message 2, if GBA authentication is needed and if there are no available bootstrapping parameters in the UE, the NAF sends a Bootstrapping initiation request.

In message 3, the UE contacts the BSF by sending an HTTP request including the private user identity.

In message 4 and 5, the BSF contacts the HSS to get the GBA User Security Setting (GUSS) and the Authentication Vector (AV) which includes RAND, AUTN, CK, IK and XRES.

In message 6 and 7, in order to authenticate the UE, the BSF sends the RAND and AUTN to the UE in a "401 (Unauthorized) message" without delivering CK, IK and XRES. The UE then verifies AUTN (following the same procedure described in IMS authentication) to conclude if the request is from an authorized network; and calculates CK, IK and RES. Thus, the session keys IK and CK are shared by the BSF and the UE. The latter generates key material Ks by concatenating CK and IK and sends the Digest AKA response which is calculated using RES in a new HTTTP request to the BSF.

In message 8, the BSF verifies the Digest AKA response to authenticate the UE and generates key material Ks by concatenating CK and IK. The BSF also generates the Bootstrapping Transaction Identifier (B-TID) and sends it in a "200 OK message" to the UE to indicate the authentication success.

The procedure to derive key material Ks_NAF is described in [33.220 09] as “Ks_NAF is computed as Ks_NAF = KDF (Ks, "gba-me", RAND, IMPI, NAF_Id), where KDF is the Key Derivation Function, and the key derivation parameters consist of the user’s IMPI, the NAF_Id and RAND. The NAF_Id is constructed as follows:

NAF_Id = FQDN of the NAF || Ua security protocol identifier

(FQDN is the Fully Qualified Domain Name)”. Ks_NAF shall be used for securing the reference point Ua.

In message 9, the UE contacts the NAF and provides the B-TID and a digest calculated using Ks_NAF.

In message 10 and 11, the NAF requests the corresponding Ks_NAF and GUSS from the BSF by sending B-TID over Zn. After receiving the BSF response (containing Ks_NAF, GUSS and the PKG parameters), the NAF calculates the digest values using Ks_NAF and compares the calculated values with the received one to be able to authenticate the UE.

At the end, the NAF sends the message 12 including an “OK message” to end the GBA authentication.

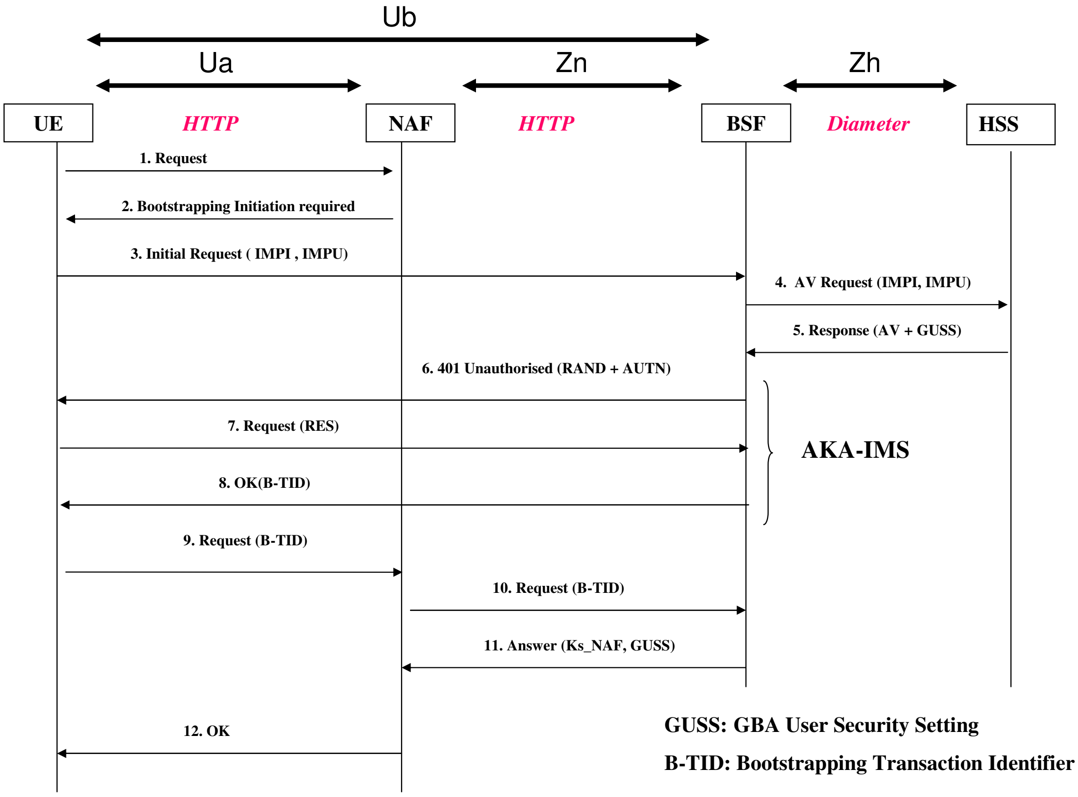

Ub
Ua
Zn
Zh
UE
HTTP
NAF
HTTP
BSF
Diameter
HSS
1. Request
2. Bootstrapping Initiation required
3. Initial Request (IMPI , IMPU)
4. AV Request (IMPI, IMPU)
5. Response (AV + GUSS)
6. 401 Unauthorised (RAND + AUTN)
7. Request (RES)
AKA-IMS
8. OK(B-TID)
9. Request (B-TID)
10. Request (B-TID)
11. Answer (Ks_NAF, GUSS)
12. OK
GUSS: GBA User Security Setting
B-TID: Bootstrapping Transaction Identifier

Figure 6.5: GBA Authentication for Services Access in IMS

6.3 IMS-IBC Service Authentication

This section presents our proposed solution which employs IBC in the IMS GBA authentication. This solution was designed in a collaboration work with a team of Orange Labs Issy-les-Moulinaux [Abid 09]. We were more concentrating on security and the use of the Identity Based Cryptography (IBC). Also, the security analysis and the formal analysis using the Automatic Validation of Internet Protocols and Application (AVISPA) was done in our department. Our objective is to personalize the IMS service authentication process through using an identity that is not attached to the ISIM card and that presents each user in an individual manner.

In our work, we assume that the UE has the shared key *sk* with HSS. The HSS is located in the same server with the HSS. The PKG's public parameters (the elliptic curve

E, prime number p, order q, point P, point P_{pub} and MapToPoint: Hash function) stored in the ISIM card. The PKG has a secret private key S.

6.3.1 Solution Description

The novel IMS-IBC Service Authentication is illustrated in Figure 6.6.

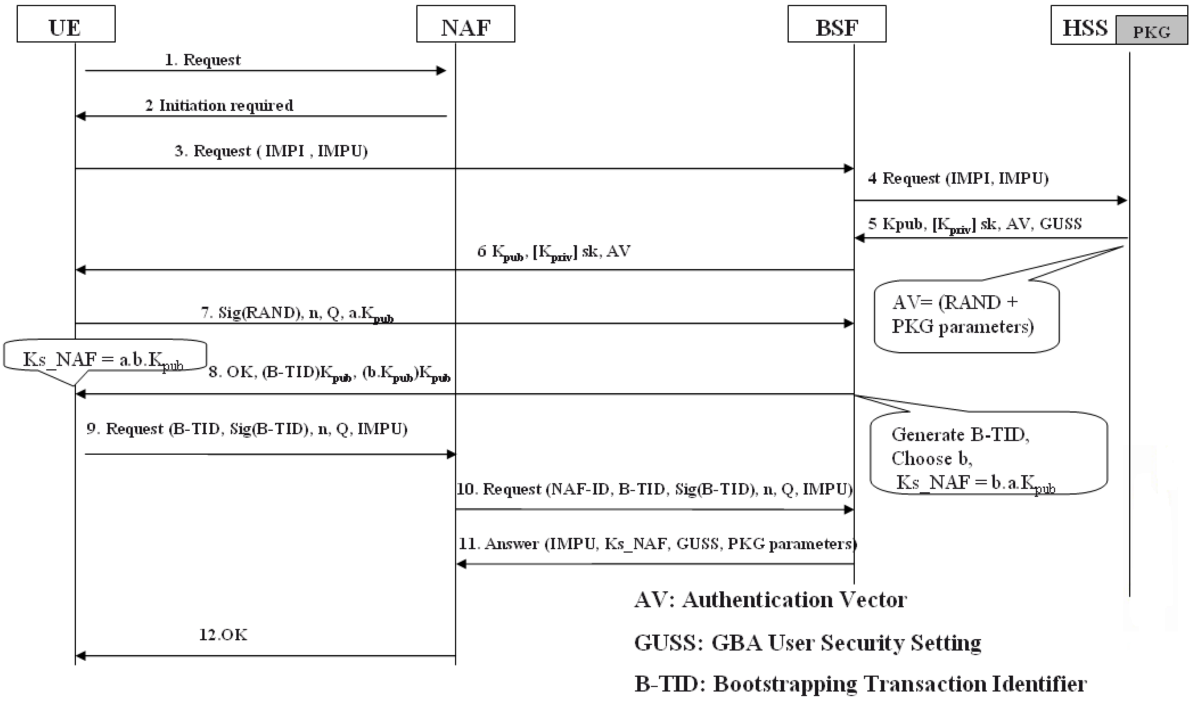

UE
NAF
BSF
HSS
PKG
1. Request
2 Initiation required
3. Request (IMPI , IMPU)
4 Request (IMPI, IMPU)
5 Kpub, [K_{priv}] sk, AV, GUSS
6 K_{pub}, [K_{priv}] sk, AV
AV= (RAND + PKG parameters)
7. Sig(RAND), n, Q, a.K_{pub}
Ks_NAF = a.b.K_{pub}
8. OK, (B-TID)K_{pub}, (b.K_{pub})K_{pub}
Generate B-TID, Choose b, Ks_NAF = b.a.K_{pub}
9. Request (B-TID, Sig(B-TID), n, Q, IMPU)
10. Request (NAF-ID, B-TID, Sig(B-TID), n, Q, IMPU)
11. Answer (IMPU, Ks_NAF, GUSS, PKG parameters)
12.OK
AV: Authentication Vector
GUSS: GBA User Security Setting
B-TID: Bootstrapping Transaction Identifier

Figure 6.6: IMS-IBC Service Authentication

We explain all the IMS-IBC interactions in the following:

In messages 1 and 2, the UE starts communication with the NAF without GBA parameters. If the NAF requires the use of shared keys obtained by means of the GBA, it replies with a bootstrapping initiation message.

In messages 3, 4 and 5, the UE sends a HTTP request to the BSF including the IMS private user identity (IMPI) and public user identity (IMPU). The BSF then retrieves from the HSS:

1. the complete set of GBA user security settings (GUSS),
2. an Authentication Vector (AV) containing the RAND and PKG parameters,
3. the UE's public key K_{pub} = MapToPoint(IMPU) where MapToPoint is the hash function used by the PKG to convert a string into a point in the elliptic curve E,
4. the encrypted UE's private key K_{priv} using the shared key sk. We note that $K_{priv} = S.K_{pub}$.

In message 6, the BSF forwards the AV (RAND and PKG parameters) and the encrypted K_{priv} to the UE in the "401 (Unauthorized) message". We add also the public key K_{pub}. We choose to send PKG parameters to the UE as HSS periodically change its parameters.

In message 7, the UE extracts its private key K_{priv} using the shared key sk (stored in the ISIM card). Then, it generates a signature of the RAND value (message m = RAND in this work) using Elliptic Curve Digital Signature Algorithm (ECDSA) (see Section 2.6.3.3). The inputs of the ECDSA are:

- PKG parameters (E(A, B), p, q)
- UE's public/private key pair (K_{pub}, K_{priv})
- n: large prime which divide the number of points in the elliptic curve
- d = K_{priv} mod (n-2)
- Q = d. K_{pub}

After the execution of the *ECDSA Signature Generation* phase, the signature for the value RAND is the pair of integers *(r, s)*.

Then, we apply Elliptic Curve Diffie-Hellman (ECDH) Protocol (presented in Section 2.6.3.1). This key agreement protocol is used to generate the Ks_NAF key. The UE chooses a random value 'a' to generate '$a.K_{pub}$'. It sends Sig(RAND) = *(r, s)*; n; Q and '$a.K_{pub}$' to the BSF in an HTTP request in order to authenticate itself.

To verify the UE's signature *(r, s)* for RAND, the BSF should follow the step of *ECDSA Signature Verification* phase. If the verification phase is successful, then, the user is authenticated.

In message 8, after the successful verification, the BSF generates Bootstrapping Transaction IDentifier (B-TID) and stores it with the IMPU and GUSS. The BSF generates a random value 'b' and sends '$b.K_{pub}$' to the UE . After receiving the message, the UE and the BSF share the same Ks_NAF = $a.b.K_{pub}$. The BSF then sends to the UE a "200 OK message" including the B-TID and '$b.K_{pub}$' encrypted with UE's public key K_{pub} using the Meneze-vanstone protocol(see Section 2.6.3.2). After receiving the message, the UE retrieves the B-TID using K_{priv}.

In message 9, the UE provides IMPU, B-TID, a signature of B-TID and n, Q to the NAF to allow it to retrieve the corresponding keys from the BSF.

In message 10, the NAF sends IMPU, NAF-ID, the signature value and n, Q to the BSF to request for Ks_NAF, GUSS and PKG parameters. NAF-ID is used by the BSF to verify that the NAF is authorized to use that hostname.

In message 11, the BSF verifies the signature using K_{pub}. Then, it sends the GUSS, KS_NAF, IMPU and PKG parameters to the NAF.

In message 12, the NAF checks the authentication and the authorization of the IMPU to the services according to the received GUSS. Once the execution of the protocol is completed, the UE and the NAF communicate in a secure way.

6.3.2 Advantages of the proposed solution

The advantages of our proposed solution are as follows:

1. Using the AKA protocol only in the IMS authentication phase while using the Identity Based asymmetric encryption for the service authentication. Thus, we can have personalized services authentication, which could not be provided by the AKA approach only. This is so important for promoting new services involving an authentication to a third party that is neither a provider nor a subscriber and that could not be easily extended with the legacy solution.

2. The session key Ks_NAF is only shared by the UE and the NAF, where the BSF only had the role to authenticate the UE. As a result, our solution is more scalable (especially when considering scenarios having several NAFs) through reducing the overhead of the key request from BSF by the NAF in the classical case.

3. Our proposed solution provides more simplicity since it is based on Elliptic Curve Cryptography (ECC), so it is more efficient and preferable in the applications that require low memory and rapid transaction. Furthermore, for elliptic-curve-based protocols, it is assumed that finding the discrete logarithm of an elliptic curve element is infeasible. The size of the elliptic curve determines the difficulty of the problem.

4. Our proposed solution is compatible with the IMS standard architecture which facilitates its deployment.

5. The work is based on asymmetric cryptography, where the shared key ks between

the UE and the HSS is used to encrypt the UE's K_{priv}. and the BSF cannot retrieve this key and has to encrypt B-TID using UE's K_{pub}. Then, even if there is no mutual authentication between the UE and the BSF as in the classical IMS case, security is always guaranteed.

6.4 Security Analysis

We explain, first, how our solution enhances the security of the classical IMS authentication through overcoming two possible attack scenarios. Then, we present the result of the formal security analysis using the Automatic Validation of Internet Protocols and Application (AVISPA).

6.4.1 Preliminary Discussion

Our solution prevents the illegitimate use of IMS services with stolen credentials is a result of UE based frauds.

6.4.1.1 Eavesdropping Attack

Through the authentication phase (messages 3 to 8), a malicious user can act as a Man in the Middle (MITM), listens to the communication and retrieves the GUSS related to the IMPI and IMPU of the legitimate user. The malicious user then tries to connect to the system using his/her ISIM card (containing IMPI' and IMPU). The HSS rejects the request because the couple (IMPI', IMPU) is not in its database.

Even, if the malicious user would play the role of a BSF, he cannot retrieve K_{priv}, because it is encrypted using sk shared between the UE and the HSS.

6.4.1.2 Impersonating UE

A malicious UE could impersonate the legitimate UE to have access to its services. When a malicious UE knows IMPI and IMPU, he/she can create a fake ISIM card containing IMPI and IMPU. When the malicious UE receives message 6, he/she only extracts K_{pub} and AV, however he/she can not recover the K_{priv} value because he/she does not know the shared key ks. Then, when he/she needs to sign RAND, the result will be a non valid signature.

6.4.2 AVISPA Automatic Validation Tool

In order to analyze the security of the proposed solution, we used the Automatic Validation of Internet Protocols and Application (AVISPA) tool, a comprehensive security protocol analyzer [AVISPA 03]. AVISPA uses the High Level Protocol Specification Language (HLPSL) [Chevalier 04] to describe security protocols and specify which security goals are achieved by a given protocol. System behavior in HLPSL is modeled as a state. The HLPSL has states which have variables responsible for the state transitions; if variables change, a state takes a new form. The communicating entities are called roles. The role's variables can be local or global. The roles are the initiators, the receivers, the environment and the session of protocol execution. Roles can have one agent or more. All communications are synchronous between roles and the intruder. These communication are carried by communication channels depending of the environment. Different intruder models of various attack capabilities can be tested using the different security properties of each communication channel. In HLPSL, the most employed intruder model is Dolev-Yao [Dolev 83] in which the following assumptions hold:

– The intruder is able to impersonate any user.
– The intruder is able to read, suppress and modify all messages exchanged between the legitimate participants in the network.
– The intruder is able to generate new messages at any time, send them to all the agents and also play the role of any legitimate principal, taking a part of the session or even multiple sessions of the protocol execution.

In our study, we choose the Dolev-Yao intruder model.

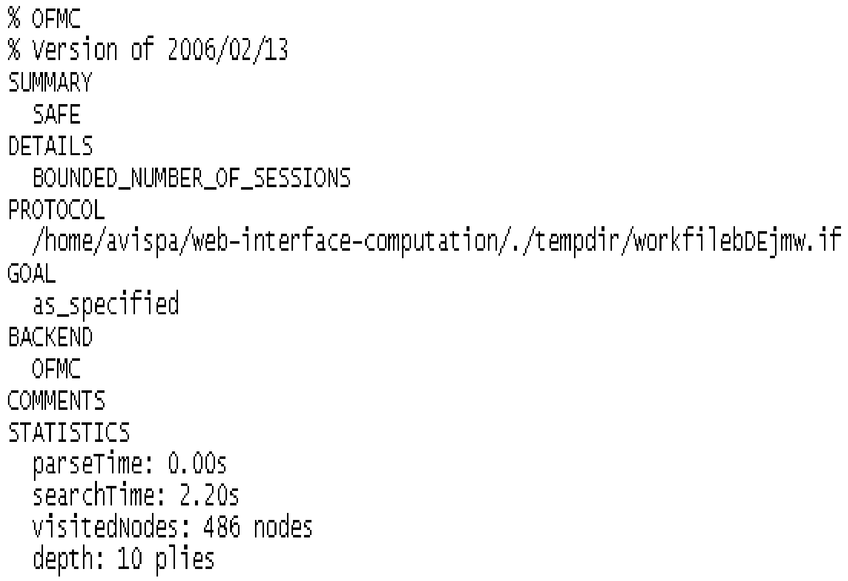
After the protocol is written, it is first compiled by the HLPSL translator into Intermediate Format (IF). The IF is a low-level language that is understood by the four back end analysis tools of AVISPA. The execution of a protocol written in IF is done in a finite number of iterations, or entirely in case of no loop is involved. The result can be either an attack is found, or the protocol is safe over the given number of sessions. We have used On-the-fly-Model-Checker (OFMC) [Basin 03] tool since it provides support for specific algebraic properties, in our case the exponential operator used for Diffie-Hellman construct (we used exponential operator to simulate the multiplication operator in the elliptic curve field.)

6.4.3 Validation Results

We defined the security goals as an input to the automatic formal proof of the IMS-IBC Service Authentication Protocol:

- Secrecy of the session key ks between the UE and the HSS.
- Secrecy of the TLS session in the reference point Ub.
- Secrecy of the DIAMETER connection in the reference point Zh.
- Secrecy of the material key Ks_NAF.
- Authentication of UE by the BSF.
- Even if an intruder impersonate the UE to the BSF, this should not allow him to get the Ks_NAF and access to the services.

The output shows that the protocol is safe (no attacks were found) and that the security goals of our formal validation are attained. The output of OFMC is shown in figure 6.7. And, the HLPSL code is given in Appendix D.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfilebDEjmw.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 2.20s
  visitedNodes: 486 nodes
  depth: 10 plies
```

Figure 6.7: The OFCM Output

We add some details about our validation. The out-of-band channel was modeled using a secret, shared between the UE and the HSS, and unknown to the intruder. As AVISPA do not have any elliptic curve algebraic functions, we simulated the multiplication function in the $\mathbb{F}_p$ by the exponential function in $\mathbb{R}$. In fact, they are analogous in their functionalities. Also the ECDSA signature is simulated by the hash_func defined in AVISPA. We correctly compiled HLPSL model and validated the IMS-IBC Service Authentication Protocol.

6.5 Performance Evaluation

Our goal is to have a first evaluation on the feasibility of the proposed solution before integrating it in an IMS platform. We separated performance evaluations for the proposed authentication approach and measured the speed of IBE's operations “the generation of PKG parameters and private and public keys” using ordinary terminals (that are expected to be used in the daily life by any user). We believe that IBE performance is the most critical point that can judge the feasibility of deploying our solution but the measures that we got are encouraging.

More precisely, the most important parameter influencing the performance is the speed of cryptographic operations (such as private/public key pair generation, encryption/decryption, and signature/verification time). In the performance analysis in this section, we mainly measure the speed of cryptographic operations, without considering the underlying network architecture at this phase. We notice the existence of IMS platforms [OIC]. However, without any implementation of the GBA authentication for services access.

Mkwawa et al. [Mkwawa 08] made some simulations on IMS and found that the S-CSCF is the bottleneck in the IMS core network because this latter processes large number of SIP messages. Analogically, we estimate that the BSF is the bottleneck of the service authentication process since it processes many HTTP messages.

We implemented the architecture presented in Figure 6.8. We have implemented four processes playing the role of the User Equipment (UE), the Network Application Function (NAF), the Bootstrapping Server Function (BSF) and the Home Subscriber Server (HSS). We added a Private Key Generator (PKG) in the HSS.

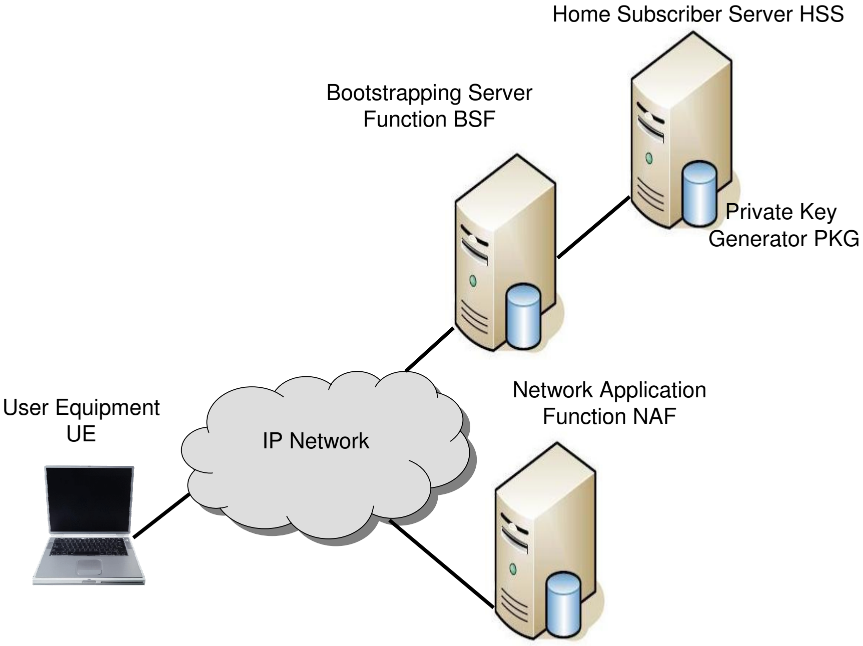

Home Subscriber Server HSS
Bootstrapping Server Function BSF
Private Key Generator PKG
User Equipment UE
IP Network
Network Application Function NAF

Figure 6.8: Testing Architecture

In order to evaluate the performance of our solution, we used the IBE demo provided within the MIRACL library [MIRACL], to determine the processing time to generate the private/public key pair and encryption/decryption. We use the elliptic curve $y^2 = x^3 + 1$ mod p where p is 256-bit prime number.

We observed the execution time for an UE who wants to be authenticated for the first time. The result we obtain is around 250 ms using a computer machine with this configuration: Intel Centrino Duo P8700 2.53GHz, memory 2G in Linux Fedora12. Moreover, we observed the execution time of some sub-functions, like:

1. Time needed to generate the PKG parameters (160-bit q, 256-bit p, 512-bit point P, 512-bit Point P_{pub}, 160-bit secret S and 512-bit cube root of unity in F_p^2), (ibe_ext) for a client is around 26.3 ms.
2. Time needed to generate and verify a digital signature (functions *ecsign* and *escver*) are nearly 9.3 ms and 10.4 ms.
3. Time needed to encrypt and decrypt the private key (functions *aes_ enc* and *aes_ dec*) are around 5 ms and 5.1 ms.

4. Time needed to encrypt and decrypt the B-TID (functions *menezes_ enc* and *menezes_ dec*) are around 8.1 ms and 8.2 ms.
5. Time to calculate the product of a random number and a given point (function *multi_ a*) is nearly 4.8 ms.
6. Time to calculate the product of a given number and a given point (function *multi_ b*) is nearly 5.8 ms.
7. Converting the message between text and html needs nearly 2.7 ms.

From the performance point of view, the time needed to finish the asymmetric system operations is not harmful to our work. Furthermore, from the security point of view, the identity Based Cryptography IBC is more secure (using 160 key instead of 128 key for AKA).

6.6 Use of Batch Verification Scheme

In Section 6.5, we validated our proposal for one user. Since the Operator has a large number of subscribers (in some cases, millions of them), we focus in this section on the the case of multi-authentication at the same time. We think that the bottleneck of the IMS's service authentication is the Bootstrapping Server Function (BSF) because it has the role to verify the signature of each User Equipement (UE). One solution to this problem is the Batch Verification scheme which is used by Zhang et al. [Zhang 08]. Their solution is called Identity-based Batch Verification (IBV) scheme and it is applied to Vehicular Network. The IBV has four phases: Key Generation and Pre-distribution, Pseudo Identity Generation, Message Signing and Batch Verification. The authors define the Batch Verification scheme as "the verification of all the signatures received in a time window with rather short time compared to verifying each signature one after the other". They indicated that the batch cryptography based on RSA was firstly proposed by Fiat [Fiat 89] in 1989. Some other batch signature schemes were suggested later like [Camenisch 07].

Zhang et al. [Zhang 08] used 3 pairing operations to verify a single signature. To verify n signatures, they needed 3 pairing operations instead of $3n$ pairing operations. In other words, the verification time of the dominant operation (i.e; pairing) is independent of the number of signatures to verify. As a result, the time spent on verifying a large number of signatures is decreased. We use their scheme in the IMS architecture.

We present the third and fourth steps of Zhang et al.'s solution and we present the method to reutilize them in the IMS's service mechanism in Appendix E.

To validate the performance of this signature method, we implemented the cryptographic operation in our proposed solution including the IBC procedures. We observe that the use of asymmetric cryptographic procedures leads to longer running time than symmetric procedures. However, the Batch Verification helps the BSF to verify the UEs signature in a reasonable time.

We made some performance evaluations and we considered the same implementation described in Section 6.5. In the following, all the measures are real measures from the implementation realized using an Intel(R) Core 2 CPU T5470 @ 1,60GHZ. We observed that the time needed to generate PKG parameters (160-bit q, 256-bit p, 512-bit point P, 512-bit Point $Ppub_1$, 160-bit secret S and 512-bit cube root of unity in F_p^2) is around 14 ms. To generate $Kpub_1$ and $Kpub_2$, we use a MapToPoint function, which has the role of finding a point in the curve E corresponding to the Hash of the IMPU. We found that the time needed for MapToPoint T_{mtp} is in the order of 4,4 ms. To generate $Kpriv_1$ and $Kpriv_2$, the PKG needs almost 7,5ms. The time needed for bilinear pairing T_{bp} is about 9,3 ms and the time for multiplication T_{mul} is about 1,5 ms.

We define the time needed for the signature verification based on the definition given [Zhang 08]. For the verification of Sig$_1$ or Sig$_2$, we need 3 bilinear pairings, 1 MapToPoint and 1 multiplication, so for 1 person $T_v = 3.T_{bp} + T_{mtp} + T_{mul} =$ 33,8 ms. The BSF has a maximum capacity, we note as N. If we have more than N UEs simultaneously requesting authentication, the system rejects or delay the answer. With the Batch Verification, this can be avoided since the verification for n signature costs three bilinear pairings, n MapToPoint and n multiplication ($T_v = 3.T_{bp} + n.T_{mtp} + n.T_{mul}$). Table 6.1 shows the time needed that we deduced for different number of UEs (we choose N >=1000 UEs).

From the performance point of view, the asymmetric system seems to need more time to finish all operations than symmetric one but this is not harmful to our work. Furthermore, from the security point of view, the identity Based Cryptography IBC is more secure (using 160 key instead of 128 key for AKA) and we calculated that we need about 4 min to authenticate 50000 UEs, and it seams to be an encouraging result.

Scenario **UEs Number**	**without Batch Scheme**	**with Batch Scheme**
1000	33,8 s	$\sim$ 5,9 s
5000	169 s	$\sim$ 29,5 s
10000	338 s	$\sim$ 59 s
50000	1690 s	$\sim$ 295 s

Table 6.1: Sinature Verification Time

6.7 Conclusion

IP Multimedia Subsystem (IMS) is promising in future services convergence and pervasive multimedia applications. Although IMS enhances user's interactivity and although security is a critical aspect, IMS authentication falls short to be realized in a personalized manner, which is an important prerequisite in new services such as social Internet ones. In this chapter, we proposed a new IMS Service Authentication scheme through employing the Identity Based Cryptography (IBC). IBC was chosen since our objective is to allow IMS services' personalization through authenticating users in a personal manner during services access. Our proposed solution also employs Elliptic Curve Cryptography (ECC), so it is more efficient and preferable in the applications that require low memory and rapid transaction. Security is assured thanks to using a symmetric protocol with a shared key (ks) between the UE and the HSS, an asymmetric protocol for signature, and Diffie-Hellman for key agreement. We focused on the eavesdropping and impersonation attacks that can take place in classical IMS scenario and we showed how our proposed solution can prevent against these attacks.

Regarding the performance of our proposed solution, we focused on the performance of the cryptographic functions in order to verify the validity of our approach. We observed that the use of asymmetric cryptographic procedures leaded to longer execution time than symmetric procedures. And, on the other hand, we have the advantage of resolving the AKA security weakness in classical IMS case.

Chapter 7

Conclusion

In this dissertation, we described new authentication protocols in different network types like Home, Governmental and Cellular Network. In this final chapter, we summarize our research contributions and briefly describe some areas that can merit future research.

7.1 Research Summary

We present three solutions , depending on the identity used in the authentication mechanisms. In chapter 4 and chapter 5, the biometric identity is used in two scenarios, one for Home Network, and the other for Governmental Network (travel document).

For the home network, we applied a biometric authentication since it allows users' authentication in a distinguished manner as well as personalized users' access. The mechanism is a modified version of the Extensible Authentication Protocol (EAP). We proposed a solution that protects private biometric template (BT) thanks to cancealable biometrics concept. To share a secret key, the entity use the fuzzy vault mechanism. This technology should be carefully used in order to protect users' privacy and prevent the disclosure of their BT.

For the the travel document case study, we proposed a new authentication protocol for e-Passport based on fingerprint. A multi-bit information string is extracted from biometrics and that string is used to generate the security parameters of the cryptographic protocol. One requirement of this system is that the minutiae points are ordered in the same way in the enrollment and verification phase, which is not easy to obtain in real circumstances. The solution has three phases where the second one is the same as defined in the Pasupathinathan et al. [Pasupathinathan 08b] except that we used Elliptic

Curve Diffie-Hellman (ECDH) Key Agreement protocol. Then, we presented an iris based authentication mechanism for e-Passport. It is a modified version of the first scheme. We obtained a key using the iris code and this key is used to generate the security parameters. We presented a security analysis and a biometric performance evaluation. We performed tests on the NIST-ICE database of iris images to compute the False Rejection Rate and the False Acceptance Rate. The results obtained (e.g., FRR of 0.2% and FAR of 3.6%) are satisfying and the use of iris biometrics is encouraging for the deployment of this solution.

In chapter 6, we used a simple strings (exp: email address,...) as an identity. The latter is used in the Identity Based Cryptography (IBC). We defined a new service authentication protocol in IP Multimedia Subsystem (IMS). The IMS is an overlay architecture for the provision of multimedia services (such as Voice over IP (VoIP), video conferencing, presence, push-to-talk, etc..) on top of all IP networks as well as Next Generation Networks (NGN). This new IMS Service Authentication scheme employs the Identity Based Cryptography (IBC). IBC was chosen since our objective is to allow IMS services' personalization through authenticating users in a personal manner during services access. Security is assured thanks to using a symmetric protocol with a shared key (ks) between the UE and the HSS, an asymmetric protocol for signature, and Diffie-Hellman for key agreement. We focused on the eavesdropping and impersonation attacks that can take place in classical IMS scenario and we showed how our proposed solution can prevent against these attacks. We, then, proposed to add a Batch Verification on the Bootstrapping Server Function (BSF) to decrease signature verification delay and the authentication response time. To validate the performance of our proposed solution, we implemented the cryptographic operation in our proposed solution including the IBC procedures. We observed that the use of asymmetric cryptographic procedures leads to longer running time than symmetric procedures. However, the Batch Verification helps the BSF to verify the User Equipments (UEs) signature in a reasonable time.

7.2 Future Work

The authentication protocols are vital to access services in the Network. We present here three possible future works:

1. A problem related to user privacy will rise which consists on the way to protect the privacy of the user. Sometimes the identity can be biometric data which is sensitive to attack and forgery. In other cases, the users of the network want to be

anonymous. So we need to develop a trust model so the users can trust the providers and the privacy will be preserved. A privacy-based enhanced access control should be more explored and extended to support privacy preferences.

2. Another issue will be related to biometrics, in our work, we focused on a single biometric data like fingerprint or iris code. As future work, we want to derive algorithm for multimodal biometric templates (see Section 3.3) since the e-Passport can contain different types of biometrics such as fingerprint, face, iris, etc.

3. Another future work can be a contribution in the electronic Visa (e-Visa) security protocol. In order to protect their borders and enforce immigration policies, Europe and in particular the Schengen countries, plan to implement a secured visa. The chosen option is a microprocessor-based solution, e-Visa, and pilots are taking place to test the technology and its implementation. What kind of Biometric Data should be stored within E-visa? How will be the interaction between e-visa and the Inspection System (IS)? Another interesting field is the e-traveler's check [Chang 09], to make shopping when traveling, easier and more secure.

Bibliography

[33.102 09] 3PPP TS. 33.102. *3G security. Security Architecture(Release 8)*. In 3GPP Technical Specification TS 33.102, 2009.

[33.203 09] 3PPP TS. 33.203. *3G security. Access security for IP-based services(Release 9)*. 2009.

[33.210 09] 3PPP TS. 33.210. *3G security. Network Domain Security (NDS). IP network layer security(Release 9)*. 2009.

[33.220 09] 3PPP TS. 33.220. *Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture(Release 9)*. 2009.

[33.228 09] 3PPP TS. 33.228. *IP Multimedia Subsystem (IMS); Stage 2(Release 9)*. 2009.

[3GPP] 3GPP. *3rd Generation Partnership Project*. In http://www.3gpp.org/.

[Abid 09] M. Abid, S. Song, H. Moustafa & H. Afifi. *Efficient Identity-Based Authentication for IMS based Services Access*. In the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM2009), pages 278–284, 2009.

[Abid 10] M. Abid, S. Kanade, D. Petrovska-Delacrétaz, B. Dorizzi & H. Afifi. *Iris Based Authentication Mechanism for e-Passports*. In the 2nd International Workshop on Security and Communication Networks (IWSCN 2010), pages 77–81, 2010.

[Aboba 04] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson & H. Levkowetz. *Extensible Authentication Protocol (EAP)*. In RFC 3748, 2004.

[Ahmad 09] A. Ahmad, A. Biri, H. Afifi & D. Zeghlache. *TIBC : Trade-off between Identity-Based and Certificateless Cryptography for future Internet*. In PIMRC'09, pages 2866–2870, 2009.

[Al-Riyami 03] S. S. Al-Riyami & K. G. Paterson. *Certificateless public key cryptography.* In ASIACRYPT,volume 2894 of Lecture Notes in Computer Science, pages 452–473, 2003.

[AVISPA 03] AVISPA. *Automated validation of internet security protocols and applications.* In FET Open Project IST-2001-39252. www.avispa-project.org., 2003.

[Basin 03] D. Basin, S. Mdersheim & L. Vigan. *An On-the-Fly Model-Checker for Security Protocol Analysis.* In in proceedings of ESORICS 2003, Lecture Notes in Computer Science, Volume 2808, pages 253–270, 2003.

[Batina 06] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls & I. Verbauwhede. *An Elliptic Curve Processor Suitable For RFID-Tags.* In Availble on: http://eprint.iacr.org/2006/227.pdf, 2006.

[Bellare 96] M. Bellare, R. Canetti & H. Krawczyk. *Keying Hash Functions for Message Authentication.* In Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, pages 1–15, 1996.

[Bersani 04] F. Bersani. *EAP shared key methods: a tentative synthesis of those proposed so far.* In draftbersani-eap-synthesis-sharedkeymethods-00, 2004.

[BlueKrypt 10] BlueKrypt. *Cryptographic Key length Recommendation.* In http://www.keylength.com/, 2010.

[Bolle 02] R. M. Bolle, J. H. Connell & N. K. Ratha. *Biometric perils and patches.* In Pattern Recognition Volume 35, Issue 12, pages 2727–2738, 2002.

[Boneh 01] D. Boneh & M. Franklin. *Identity-Based Encryption from the Weil Pairing.* In Proceedings of CRYPTO '01, LNCS 2139, pages 213–229, 2001.

[Boneh 02] D. Boneh. *Voltage Security.* In http://www.voltage.com/vsn/, 2002.

[Borisov 01] N. Borisov, I. Goldberg & D. Wagner. *Intercepting mobile communications: the insecurity of 802.11.* In Proceedings of the 7th annual international conference on Mobile computing and networking MobiCom'01, pages 180–189, 2001.

[Bringer 07] J. Bringer, H. Chabanne, G. Cohen, B. Kindarji & G. Zemor. *Optimal Iris Fuzzy Sketches.* In IEEE Conference on Biometrics: Theory, Applications and Systems BTAS'07, 2007.

[Burmester 09] M. Burmester, T. Van Le, B. De Medeiros & G. Tsudik. *Universally Composable RFID Identification and Authentication Protocols.* In Transactions on Information and System Security (TISSEC), volume 12. IEEE, 2009.

[Burnett 07] A. Burnett, F. Byrne, T. Dowling & A. Duffy. *A Biometric Identity Based Signature Scheme.* International journal of network security, volume 5, issue 3, 2007.

[Calhoun 03] P. Calhoun, J. Loughney, E. Guttman, G. Zorn & J. Arkko. *Diameter Base Protocol.* In IETF RFC 3588, 2003.

[Camarillo 04] G. Camarillo & M. A. Garcia-Martin. The 3g ip multimedia subsystem (ims): merging the internet and the cellular worlds. John Wiley & Sons, 2004.

[Camenisch 07] J. Camenisch, S. Hohenberger & M. Pedersen. *Batch verification of short signatures.* In in Proceedings of EUROCRYPT, LNCS, Vol. 4514, pages 246–263, 2007.

[Cavoukian 07] A. Cavoukian & A. Stoianov. *Biometric Encryption: A Positive-Sum Technology that Achieves Strong Authentication, Security AND Privacy.* In http://whitepapers.techrepublic.com.com/ abstract.aspx?docid=1294403, 2007.

[Chang 09] C. Chang & S. Chang. *The design of e-traveler's check with efficiency and mutual authentication.* In Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, pages 309–316, 2009.

[Chevalier 04] Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drieslma, J. Mantovani, S. Mdersheim & L. Vigneron. *A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols.* In in Proceedings of Workshop on Specification and Automated Processing of Security Requirements (SAPS 2004), pages 193–205, 2004.

[CNIL 07] CNIL. *Commission Nationale de l'Informatique et des Libertés, Dossier biométrie.* In http://www.cnil.fr, 2007.

[Daugman 94] J. Daugman. *Biometric Personal Identification System Based on Iris Analysis.* In U.S. Patent No. 5,291,560, 1994.

[Davida 99] G. I. Davida, Y. Frankel, B. J. Matt & R. Peralta. *On the relation of error correction and cryptography to an offline biometric based identification scheme.* In Proceeding Workshop on Coding and Cryptography (WCC), pages 129–138, 1999.

[Dodis 04] Y. Dodis, L. Reyzin & A. Smith. *Fuzzy extractors: how to generate strong keys from biometrics and other noisy data.* In International Conference Theory and Applications of Cryptographic Techniques (EUROCRYPT) 2004, pages 523–540, 2004.

[Dolev 83] D. Dolev & A. C. Yao. *On the Security of Public Key Protocols.* In IEEE Transactions on Information Theory, 29(2), pages 198–208, 1983.

[Durlanik 05] A. Durlanik & I. Sogukpinar. *SIP authentication scheme using ECDH.* In ENFORMATIKA 1305-5313 Vol 8, pages 350–353, 2005.

[ElGamal 85] T. ElGamal. *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.* In IEEE Transactions on Information Theory, v. IT-31, n. 4, pages 469–472, 1985.

[Ellison 02] C. M. Ellison. Home network security. Intel Technology, 2002.

[EU 06] EU. *Justice and Home Affairs: Eu standard specifications for security features and biometrics in passports and travel documents.* Technical report, European Union, 2006.

[Fiat 89] A. Fiat. *Batch RSA.* In in Proceedings of Crypto, pages 175–185, 1989.

[Funk 06] P. Funk. *EAP Tunneled TLS Authentication Protocol Version 1 (EAP-TTLSv1).* In IETF draft-funk-eap-ttls-v1-01, 2006.

[Gura 04] N. Gura, A. Patel, A. Wander, H. Eberle & S. C. Shantz. *Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs.* In Cryptographic Hardware and Embedded Systems - CHES 2004, pages 119–132, 2004.

[Haller 98] N. Haller, C. Metz, P. Nesser & M. Straw. *A One-Time Password System.* In RFC 2289, 1998.

[Hankerson 04] D. Hankerson, A. Menezes & S. Vanstone. Guide to elliptic curve cryptography. Springer-Verlag New York, Inc, 2004.

[Hao 06] F. Hao, R. Anderson & J. Daugman. *Combining Crypto with Biometrics Effectively.* In IEEE Transaction Computers 55(9), pages 1081–1088, 2006.

[Hoepman 06] J.H. Hoepman, E. Hubbers, B. Jacobs, M. Oostdijk & R. W. Schreur. *Crossing Borders: Security and Privacy Issues of the European e-Passport.* In 1st Int. Workshop on Security, IWSEC 2006, LNCS 4266, pages 152-167, Kyoto, Japan, pages 152–167, 2006.

[Huang 07] C. Huang & J. Li. *One-Pass Authentication and Key Agreement Procedure in IP Multimedia Subsystem for UMTS.* In Advanced Information Networking and Applications, AINA '07, pages 482–489, 2007.

[ICAO 03] ICAO. *Machine readable travel documents. part 1. - machine readable passports.* Technical report, document 9303, part 1, ICAO, 2003.

[ICAO 06] ICAO. *Machine Readable Travel Documents. Part 1: Machine Readable Passport, Specifications for Electronically enabled Passports with Biometric Identification Capabilities.* In International Civil Aviation Organization, ICAO Doc 9303, available on: http://www2.icao.int/en/MRTD/Pages/default.aspx, 2006.

[Itakura 05] Y. Itakura & S. Tsujii. *Proposal on a multifactor biometric authentication method based on cryptosystem keys containing biometric signatures.* In Int. J. Inf. Sec. 4(4), pages 288–296, 2005.

[Jain 97] A. K. Jain, H. Lin, S. Pankanti & R. Bolle. *An identity- authentication system using fingerprints.* In Proceedings of the IEEE, Volume: 85, Issue: 9, pages 1365–1388, 1997.

[Jin 04] A. T. B. Jin, D. Ngo, C. Ling & A. Goh. *Biohashing: two factor authentication featuring fingerprint data and tokenised random number.* In Pattern Recognition, 37(11), pages 2245–2255, 2004.

[JMRTD 06] JMRTD. *A Free Implementation of Machine Readable Travel Documents.* In http://jmrtd.org/, 2006.

[Johnson 98] D. B. Johnson. *Elliptic curve DSA (ECSDA): an enhanced DSA.* In Proceedings of the 7th conference on USENIX Security Symposium - Volume 7, pages 13–23, 1998.

[Juels 99] A. Juels & M. Wattenberg. *A fuzzy commitment scheme.* In In Proceedings of the Sixth ACM Conference on Computer and communication Security (CCCS), pages 28–36, 1999.

[Juels 02] A. Juels & M. Sudan. *A fuzzy vault scheme.* In Proceeding IEEE International Symposium Information Theory, page 408. A. Lapidoth and E. Teletar, editors, 2002.

[Juels 05] A. Juels, D. Molnar & D. Wagner. *Security and Privacy Issues in E-passports.* In SecureComm' 05, pages 74–88. IEEE, 2005.

[Kanade 08] S. Kanade, D. Camara, E. Krichen, D. Petrovska-Delacrétaz & B. Dorizzi. *Three factor scheme for biometric-based cryptographic key regeneration using iris.* In In The 6th Biometrics Symposium 2008 (BSYM2008), pages 59–64, 2008.

[Kc 05] G. S. Kc & P. A. Karger. *Security and Privacy Issues inMachine Readable Travel Documents (MRTDs).* In 10th European Symposium on Research in Computer Security (ESORICS 2005) Milan, Italy, pages 14–16, 2005.

[Kim 09] G. W. Kim, D. G. Lee, J. W. Han, S. C. Kim & S. W. Kim. *Security Framework for Home Network: Authentication, Authorization, and Security Policy.* In Lecture Notes in Computer Science, Volume 4819/2009, pages 621–628, 2009.

[Koblitz 87] N. Koblitz. *Elliptic curve cryptosystems.* In Mathematics of Computation, volume 48, pages 203–209, 1987.

[Kumar 09] A. Kumar & A. Kumar. *Development of a new cryptographic construct using palmprint-based fuzzy vault.* In EURASIP Journal on Advances in Signal Processing , Volume 2009. Hindawi Publishing Corp., 2009.

[Lee 06] Y. K. Lee, H. I. Ju, D. W. Kim & J. W. Han. *Home Network Modelling and Home Network User Authentication Mechanism Using*

Biometric Information. In EEE 10th International Symposium on Consumer Electronics, ISCE2006, pages 1–5, 2006.

[Linnartz 03] J.P. Linnartz & P. Tuyls. *New shielding functions to enhance privacy and prevent misuse of biometric templates.* In Proceeding 4th International Conference Audio and Video based Biometric Person Authentication, pages 393–402, 2003.

[Logic 09] Trusted Logic & Infineon Technologies AG. *Certification Report ANSSI-2009/21 TL ICAO LDS Smart Card.* In Certification report reference ANSSI-2009/21, 2009.

[Lumini 07] A. Lumini & L. Nanni. *An improved biohashing for human authentication.* In Pattern Recognition, 40(3), pages 1057–1065, 2007.

[Menezes 93] A. Menezes & S. A. Vanstone. *Elliptic Curve Cryptosystems and Their Implementation.* In Journal of Cryptology, 6, pages 209–224, 1993.

[Menezes 97] A. Menezes, P. Van Oorschot & S. Vanstone. Handbook of applied cryptography, chapitre 10. CRC Press, 1997.

[Microsoft 10] C. Microsoft. *Introduction to the Windows Biometric Framework (WBF)- Guidelines for IHV, ISVs and OEMs.* In http://www.microsoft.com/whdc/Device/biometric/WBFIntro.mspx, 2010.

[Miller 85] V. Miller. *Use of elliptic curves in cryptography.* In Crypto'85, pages 417–426, 1985.

[MIRACL] MIRACL. *Multiprecision Integer and Rational Arithmetic C/C++ Library.* In available in http://www.shamus.ie/.

[Mkwawa 08] I. M. Mkwawa & D. D. Kouvatsos. *Performance Modelling and Evaluation of Handover Mechanism in IP Multimedia Subsystems.* In Fifth International Conference on Performance Modelling and Evaluation of Heterogeneous Networks, 2008.

[Monrose 99] F. Monrose, M. K. Reiter & S. Wetzel. *Password hardening based on keystroke dynamics.* In Proceeding 6th ACM Conference Computer and Communications Security, pages 73–82, 1999.

[Monrose 01] F. Monrose, M. K. Reiter, Q. Li & S. Wetzel. *Cryptographic key generation from voice.* In Proceeding IEEE Symposium Security and Privacy, pages 202–213, 2001.

[Nandakumar 07] K. Nandakumar, A. K. Jain & S. Pankanti. *Fingerprint-based fuzzy vault: Implementation and Performance.* In IEEE Trans. Info. Forensics & Security, vol. 2, no. 4, pages 744–757, 2007.

[Niemi 02] A. Niemi, J. Arkko & V. Torvinen. *Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA).* In RFC 3310, 2002.

[NIST] NIST. *Iris Challenge Evaluation.* In National Institute of Science and Technology (NIST), available on http://iris.nist.gov/ice.

[Nithyanand 09] R. Nithyanand. *A Survey on the Evolution of Cryptographic Protocols in ePassports.* In available on: http://eprint.iacr.org/2009/200.pdf, 2009.

[OIC] OIC. *http://www.openimscore.org/.*

[Pasupathinathan 08a] V. Pasupathinathan, J. Pieprzyk & H. Wang. *Formal analysis of icao's e-passport specification.* In Australasian Information Security Conference (AISC2008). Conferences in Research and Practice in Information Technology (CRPIT), volume 81, pages 74–88. Brankovic, L., Miller, M. , editors, 2008.

[Pasupathinathan 08b] V. Pasupathinathan, J. Pieprzyk & H. Wang. An on-line secure e-passport protocol, volume 4991/2008, pages 14–28. Springer Berlin / Heidelberg, 2008. Book Chapter in Information Security Practice and Experience.

[Poikselka 09] M. Poikselka & G. Mayer. The ims: Ip multimedia concepts and services. John Wiley & Sons, 2009.

[Priselac 08] D. Priselac & M. Mikuc. *Security risks of pre-IMS AKA access security solutions.* available at http://www.ericsson.com/hr/etk/dogadjanja/mipro_2008/1227.pdf, Ericsson, 2008.

[Qingxian 05] W. Qingxian. *The Application of Elliptic Curves Cryptography in Embedded Systems.* In Proceedings of the Second International

Conference on Embedded Software and Systems (ICESS'05), pages 527–530, 2005.

[Rahouma 09] K. H. Rahouma. *A Modified Menezes-Vanstone Elliptic Curve Multi-Keys Cryptosystem.* In the Fifth Saudi Technical Conference and Exhibation, STCEX'09, 2009.

[Ratha 07] N. K. Ratha, J. H. Chikkerur S.and Connell & R. M. Bolle. *Generating Cancelable Fingerprint Templates.* In IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(4), pages 561–572, 2007.

[Reed 60] I. S. Reed & G. Solomon. *Polynomial Codes Over Certain Finite Fields.* In Journal Society for Industrial and Applied Mathematics, Volume 8, Issue 2, pages 300–304, 1960.

[Research 00] Certicom. Research. *Standards for efficient cryptography, SEC 1: Elliptic Curve Cryptography, Version 1.0.* 2000.

[Rigney 00] C. Rigney, S. Willens, A. Rubens & W. Simpson. *Remote Authentication Dial In User Service (RADIUS).* In IETF RFC 2865, 2000.

[Ring 06] J. Ring, K. R. Choo & E. Foo. *A New Authentication Mechanism and Key Agreement Protocol for SIP Using Identity-based Cryptography.* In AusCERT2006 R&D Stream, pages 57–72, 2006.

[Ross 04] A. Ross & A. K. Jain. *Multimodal biometrics: an overview.* In 12th European Signal Processing Conference (EUSIPCO), Vienna, Austria, pages 1221–1224, 2004.

[Sahai 05] A. Sahai & B. Waters. *Fuzzy Identity-Based Encryption.* In EUROCRYPT 2005, pages 457–473, 2005.

[Schoof 95] R. Schoof. *Counting points on Elliptic Curves over Finite Fields.* In Journal de Theorie des Nombres de Bordeaux 7, pages 219–254, 1995.

[Shamir 84] A. Shamir. *Identity-Based Cryptosystems and Signature Schemes.* In Proceedings of CRYPTO 84 on Advances in cryptology, pages 47–53, 1984.

[Sher 06] M. Sher & T. Magedanz. *Secure access to IP multimedia services using generic bootstrapping architecture (GBA) for 3G & beyond* *mobile networks.* In Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks, pages 17–24, 2006.

[Sher 07] M. Sher. *Secure Service Provisioning (SSP) Framework for IP Multimedia Subsystem (IMS).* PhD thesis, Technischen Universität Berlin, Berlin, Germany, 2007.

[Snelick 03] R. Snelick, M. Indovina, J. Yen & A. Mink. *Multimodal biometrics: issues in design and testing.* In ICMI International Conference on Multimodal Interfaces, pages 68–72, 2003.

[Snelick 05] R. Snelick, U. Uludag, A. Mink, M. Indovina & A. K. Jain. *Large-Scale Evaluation of Multimodal Biometric Authentication Using State-of-the-Art Systems.* In IEEE Transactions Pattern Analysis and Machine Intelligence 27, pages 450–455, 2005.

[Soutar 99] C. Soutar, D. Roberge, S. A. Stojanov, R. Gilroy & B. V. K. V. Kumar. *Biometric encryption.* In In R. K. Nichols, editor, ICSA Guide to Cryptography. Mc- Graw Hill, chapter 22, 1999.

[S.R0086-B 05] 3PPP2. S.R0086-B. *IMS Security Framework.* 2005.

[stallings 03] W. stallings. Cryptography and network security, principles and practice. Pearson Education, Inc, 2003.

[Tirado 08] I. Tirado. *IP Multimedia Subsystem (IMS) signaling core security.* In Proceedings of the 5th annual conference on Information security curriculum development, pages 59–63, 2008.

[TISPAN] TISPAN. *Telecoms & Internet converged Services & Protocols for Advanced Networks.* In http://www.etsi.org/tispan/.

[Uludag 06] U. Uludag & A. Jain. *Securing Fingerprint Template: Fuzzy Vault with Helper Data.* In Workshop on Privacy Research In Vision (PRIV), pages 163–170. IEEE, 2006.

[US-VISIT 04] US-VISIT. In http://www.dhs.gov/files/programs/usv.shtm, 2004.

[Vaudenay 07] S. Vaudenay. *E-Passport Threats.* In Security and Privacy, volume 5, pages 61–64. IEEE, 2007.

[Wang 05a] X. Wang, Y. L. Yin & H. Yu. *Finding Collisions in the Full SHA-1.* In CRYPTO'05, pages 17–36, 2005.

[Wang 05b] X. Wang, H. Yu & Y. L. Yin. *Efficient Collision Search Attacks on SHA-0*. In CRYPTO'05, pages 1–16, 2005.

[Wu 09] L. Wu, Y. Zhang & F. Wang. *A new provably secure authentication and key agreement protocol for SIP using ECC*. In Computer Standards & Interfaces Volume 31, Issue 2, pages 286–291, 2009.

[X9.62 99] ANSI. X9.62. *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*. 1999.

[Yarlagadda 96] R. K. Yarlagadda, J. E. Hershey & R. K. R. Yarlagadda. Hadamard matrix analysis and synthesis: With applications to communications and signal/image processing. Kluwer Academic Publishers, 1996.

[Zhang 08] C. Zhang, R. Lu, X. Lin, P.H. Ho & X.S. Shen. *An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks*. In INFOCOM' 08, pages 246–250, 2008.

Appendix A

Acronyms

3GPP	Third Generation Partnership Project
AA	Active Authentication
AAA	Authentication Authorization Accounting Protocol
AES	Advanced Encryption Standard
AKA	Authentication and key Agreement
ANSI	American National Standards Institute
AS	Application Server
ATM	Automatic Teller Machine
AUTN	Authentication Token
AV	Authentication Vector
AVISPA	Automatic Validation of Internet Protocols and Application
BAC	Basic Access Control
BioID	Biometric IDentifier
BSF	Bootstrapping Server Function
BT	Biometric Template
B-TID	Bootstrapping Transaction IDentifier
CA	Certificate Authority
CA	Chip Authentication
CK	Ciphering Key
CL-PKC	Certificateless Public Key Cryptography
CNIL	Commission Nationale de l'Informatique et des Libertés
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List

CSCA	Country Signing Certificate Authorities
CSCF	Call Service Control Functions
CVCA	Country Verifying Certificate Authorities
DES	Data Encryption Standard
DH	Diffie-Hellman
DLP	Discrete Logarithm Problem
DoD	Denial of Decryption
DOS	Denial Of Service
DV	Document Verifier
EAC	Extended Access Control
EAP	Extensible Authentication Protocol
EAP-TLS	EAP-Transport Layer Security
EAP-TTLS	EAP Tunneled TLS
EEAP	Encrypted Extensible Authentication Protocol
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
EER	Equal Error Rate
e-Passport	electronic Passport
ETSI	European Telecommunications Standards Institute
e-Visa	electronic Visa
FAR	False Acceptance Rate
FRR	False Rejection Rate
FQDN	Fully Qualified Domain Name
GAR	Genuine Acceptance Rate
GBA	Generic Bootstrapping Architecture
GUSS	GBA User Security Setting
HG	Home Gateway
HLPSL	High Level Protocol Specification Language
HMAC	Hash-based Message Authentication Code
HN	Home Network
HSS	Home Subscriber System
HTTP	Hypertext Transfer Protocol

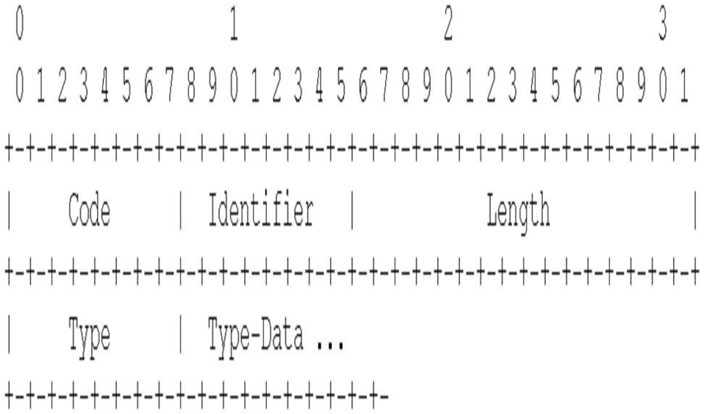
I-CSCF	Interrogating-CSCF
IBC	Identity Based Cryptography
IBE	Identity Based Encryption
IBV	Identity-based Batch Verification
ICAO	International Civil Aviation Organization
IEEE	Institute of Electrical and Electronic Engineers
IF	Intermediate Format
IFP	Integer Factorization Problem
IMPI	IP Multimedia Private Identity
IMPU	IP Multimedia Public Identity
IMS	IP Multimedia Subsystem
IK	Integrity key
IP	Internet Protocol
IS	Inspection System
ISIM	IMS-Subscriber Identity Module
KDF	Key Derivation Function
KGC	Key Generation Center
Ks_NAF	Key material of NAF
LDS	Logical Data Structure
MAC	Message Authentication Code
MIPS	Million Instructions Per Second
MIRACL	Multiprecision Integer and Rational Arithmetic C/C++ Library
MRTD	Machine-Readable Travel Documents
MRZ	Machine Readable Zone
NAF	Network Application Function
NAF-ID	Network Application Function-IDentity
NIST-ICE	National Institute of Standards and Technology-Iris Challenge Evaluation
NGN	New generation Network
NTWG	New Technologies Working Group
OFMC	On-the-fly-Model-Checker
OSEP	On-Line Secure E-passport Protocol
OTP	One Time Password

P-CSCF	Proxies-CSCF
PA	Passive Authentication
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PKD	Public Key Directory
PKG	Private Key Generator
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
RFID	Radio Frequency Identification
RSA	Rivest-Shamir-Adleman
S-CSCF	S(serving)-CSCF
SDE	Security Data Element
SEA	Schoof-Elkies-Atkin algorithm
SHA	Secure Hash Algorithm
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SOD	Document Security Object
SQN	Sequence Numbers
TA	Terminal Authentication
TISPAN	Telecoms & Internet converged Services & Protocols for Advanced Networks
TLS	Transport Layer Security
TTLS	Tunneled TLS
UE	User Equipment
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
USIM	Universal Subscriber Identity Module
VoIP	Voice over IP
WBF	Windows Biometric Framework
WEP	Wired Equivalent Privacy
XMAC	eXpected Message Authentication Code
XRES	eXpected RESult
ZKP	Zero Knowledge Proof

Appendix B

Using Fuzzy Vault in the Extensible Authentication protocol (EAP)

The Extensible Authentication Protocol (EAP), (as defined in Section 2 of the RFC 3748 [Aboba 04]) is a list of exchanged messages between an authenticator and a peer. The Request/Response packet format is presented in Figure B.1 (replicated from [Aboba 04] page 22):



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |  Type-Data ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure B.1: EAP Request Paquet Format

Each packet has these elements:

- Code: $\begin{cases} 1 & Request \\ 2 & Response \\ 3 & Success \\ 4 & Failure \end{cases}$
- Identifier: one octet
- Length: two octets, is the length, in octets, of the EAP packet (i.e, the length of the fields: Code, Identifier, Length, Type and Data-type).
- Type: $\begin{cases} 1 & Identity \\ 2 & Notification \\ 3 & Nak(Responseonly) \\ 4 & MD5-Challenge \\ 5 & OneTimePassword(OTP) \\ 6 & GenericTokenCard(GTC) \\ 254 & ExpandedTypes \\ 255 & Experimentaluse \end{cases}$
- Type-data: (a message containing at most UTF-8 encoded ISO 10646 characters) In this request, it is empty.

In our scheme, the authenticator is the Home Gateway (HG) and the peer is the User's equipement. We describe the message illustraed in the Figure 4.4.

1. The Home Gateway (HG) sends an EAP Identity Request Req1 to authenticate the user.

Req1: $\begin{cases} Code & 1 \\ Identifier & xxx \\ Length & --- \\ Type & 1 \\ Type-data & emptyfield \end{cases}$

2. The user sends an EAP Response Identity Resp1 containing :

Resp1: $\begin{cases} Code & 2 \\ Identifier & xxx \\ Length & --- \\ Type & 1 \\ Type-data & login \end{cases}$

3. The Home Gateway (HG) retrive the User's BioID using the login value. Then, it chooses a secret key and generates the vault. In the end, the HG sends a new EAP Request packet Req2 to the User. Since, we are dealing with a new secret key for each connection, our system is similar to One-Time Password (OTP) [Haller 98].

$$\text{Req2 :}\begin{cases} Code & 1 \\ Identifier & xxx+1 \\ Length & --- \\ Type & 5 for “OTP'' \\ Type-data & vault, challenge and nonce \end{cases}$$

4. The User de-blocks the vault, retreive the secret key, and encrypts the received challenge. Then, he/she sends an EAP Response packet Resp2 to the Home Gateway.

$$\text{Resp2:}\begin{cases} Code & 2 \\ Identifier & xxx+1 \\ Length & --- \\ Type & 5 for “OTP'' \\ Type-data & encrypted challenge and nonce \end{cases}$$

5. The HG continues decrypts the encrypted challenge. If it is false, it sends a EAP Failure (Code 4). In the other case it transmitsuntil an EAP Success (Code 3).

For the Vault generation and deblocking, we use the same method proposed by Uludag et al. [Uludag 06].

Appendix C

Our Proposol's Second Phase for e-Passport Authentication Mechanism

The authentication protocol is run between the chip C, the Inspection System (IS) and the Document Verifier (DV). The chip C, the IS and the DV have public/private key pair certified by the Country Verifying Certificate Authorities (CVCA).

Step 1: A traveler presents his/her e-Passport to the Inspection System (IS). The IS reads the MRZ information and sends GET CHALLENGE command to the chip C.

Step 2: The chip C generates a secret random number N_C $(1 <= N_C <= q)$ and calculates $Q_C = N_C.P$. Then, it answers to the GET CHALLENGE command by sending Q_C and the public parameters A, B, p, q and P to the IS.

Step 3: After receiving the chip C's replay, the IS chooses a random number N_{IS} $(1 <= N_C <= q)$ and computes $Q_{IS} = N_{IS} * P$. The IS creates S_{IS} by signing the message containing MRZ value and Q_C.

$$S_{IS} = SIGN\mathrm{SK}_{IS}(MRZ|Q_C)$$

The Inspection System (IS) then communicates with the traveler's DV in its proximity and obtains the DV's public key PK_{DV}. The IS encrypts S_{IS}, MRZ information and Q_C using PK_{DV}. The message contains the data encrypted with the IS's certificate signed by CVCA.

Step 4: The DV decrypts the IS's message and verifies $CERT_{CVCA}(PK_{IS}, IS)$ and the signature S_{IS}. If the verification is successful, the DV concludes about the IS's genuineness and produces a new signature S_{DV} to prove IS's authenticity to the chip C.

$$S_{DV} = SIGN\mathrm{SK}_{DV}(MRZ|Q_C|PK_{IS})$$

The DV encrypts S_{DV} the public key PK_{IS} and transfers it to the IS. The DV can send in option the chip's public key PK_C.

Step 5: The Inspection System (IS), after decrypting the received message, computes the point $Q = N_{IS} * Q_C$. As $Q = (X, Y)$, the session key can be the element X or Y. We choose $K = X$. (X has 256 bits). The IS encrypts the signature, MRZ information and Q_C using K. It also signs its point Q_{IS} and the chip's public parameters.

Step 6: The chip C computes the point $Q = N_C * Q_{IS}$. It chooses session key K like the IS did before. The chip C decrypts the received message using the session key K, retrieve the signature S_{DV} and verifies the signature $SIGN\mathrm{SK}_{IS}(Q_{IS}, A, B, p, q, P)$. The chip C is convinced about the IS's genuineness after a successful verification.

In figure C.1, we can see a resume of the interaction between the 3 entities.

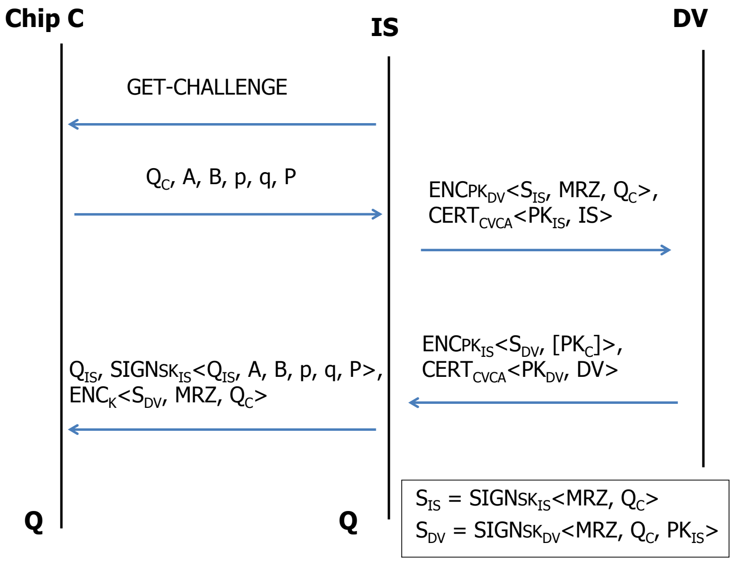

Chip C
IS
DV
GET-CHALLENGE
Q_C, A, B, p, q, P
ENCPK$_{DV}$<S$_{IS}$, MRZ, Q$_C$>,
CERT$_{CVCA}$<PK$_{IS}$, IS>
ENCPK$_{IS}$<S$_{DV}$, [PK$_C$]>,
CERT$_{CVCA}$<PK$_{DV}$, DV>
Q$_{IS}$, SIGNSK$_{IS}$<Q$_{IS}$, A, B, p, q, P>,
ENC$_K$<S$_{DV}$, MRZ, Q$_C$>
S$_{IS}$ = SIGNSK$_{IS}$<MRZ, Q$_C$>
S$_{DV}$ = SIGNSK$_{DV}$<MRZ, Q$_C$, PK$_{IS}$>
Q
Q

Figure C.1: IS Authentication

Appendix D

AVISPA

```
%%%%%%%%%%%%%%%%%%%%%% UE %%%%%%%%%%%%%%%%%
    role ue (
U,N,B,H             : agent,
    Ksnaf             : symmetric_key,  % ksnaf
Kuh                 : symmetric_key,     %sk
Kpub_u             : public_key,
    Sign               : hash_func,
          Request,
Initiation_Required,Success     : text,
SND_UN, RCV_UN, SND_UB, RCV_UB  : channel(dy))

  played_by U def=

           local  State , N1, A1, B1      : nat,
       IMPI, IMPU, RAND, B_TID  : text,
       Key_U, Key_B             : message,
       Pkg_param, Q         : nat set

           const  sec_u_kuh  : protocol_id,
       sec_u_ksnaf  : protocol_id,
       auth_R  : protocol_id,
       auth_B  : protocol_id
```

```
        init State := 0

  transition

       1. State  = 0 /\ RCV_UN(start) =|>
          State':= 1 /\ SND_UN(Request)

       2. State  = 1 /\ RCV_UN(Initiation_Required) =|>
          State':= 2 /\ IMPI':= new()
         /\ IMPU':= new()
    /\ SND_UB(IMPI'.IMPU')

       3. State  = 2 /\ RCV_UB(Kpub_u'.
         {inv(Kpub_u)}_Kuh.RAND'.Pkg_param') =|>
          State':= 3 /\ N1':=new()
    /\ Q' :=new()
    /\ A1' :=new()
    /\ SND_UB(Sign(RAND',N1',Q').N1'.Q'.exp(A1',Kpub_u'))
    /\ secret(Kuh,sec_u_kuh,{U,H})
    /\ witness(U,B,auth_R,Sign(RAND',N1',Q'))
    /\ request(U,B,auth_R,Sign(RAND',N1',Q'))

       4. State  = 3 /\ RCV_UB({B_TID'.Key_B'}_Kpub_u) =|>
          State':= 4 /\ Ksnaf' := exp(A1, Key_B')
    /\ N1':=new()
    /\ Q' :=new()
    /\ SND_UN(B_TID'.Sign(B_TID',N1',Q').N1'.Q'.IMPU)
    /\ witness(U,B,auth_B,Sign(B_TID',N1',Q'))
    /\ request(U,B,auth_B,Sign(B_TID',N1',Q'))
    /\ secret(Ksnaf',sec_u_ksnaf,{U,B})

   5. State  = 4 /\ RCV_UN(Success) =|>
          State':= 5

end role
```

```
%%%%%%%%%%%%%%%%%%%% NAF %%%%%%%%%%%%%%%%

   role naf (
U,N,B              : agent,
Ksnaf      : symmetric_key,
Request,
Initiation_Required,
Success  : text,
SND_NU, RCV_NU, SND_NB, RCV_NB : channel(dy))

 played_by N def=

           local  State, N1           : nat,
       Knb     : symmetric_key,
       IMPI, IMPU,RAND,B_TID, NAF_ID, GUSS : text,
       Pkg_param, Q : nat set,
       X : text

           const  sec_n_knb  : protocol_id,
       sec_n_ksnaf    : protocol_id,
       auth_R          : protocol_id,
       auth_B          : protocol_id

           init State := 11

    transition

            1. State  = 11 /\ RCV_NU(Request) =|>
               State':= 12 /\ SND_NU(Initiation_Required)

            2. State  = 12 /\ RCV_NU(B_TID'.X'.N1'.Q'.IMPU') =|>
               State':= 13 /\ NAF_ID':=new()
                           /\ SND_NB({IMPU'.NAF_ID'.X'.N1'.Q'}_Knb)
         /\ secret(Knb,sec_n_knb,{B,N})
```

```
            3. State  = 13 /\ RCV_NB({IMPU'.Ksnaf'.GUSS'.Pkg_param'}_Knb)
             =|>
               State':= 14 /\ SND_NU(Success)
       /\ secret(Ksnaf',sec_n_ksnaf,{B,N})
       /\ secret(Knb,sec_n_knb,{B,N})

    end role

%%%%%%%%%%%%%%%%%%%%%% BSF %%%%%%%%%%%%%%%%%

    role bsf (
U,N,B,H        : agent,
          Ksnaf     : symmetric_key,
Kpub_u         : public_key,
Sign        : hash_func,
SND_BU, RCV_BU , SND_BN, RCV_BN, SND_BH, RCV_BH : channel(dy))
    played_by B def=

         local  State, N1, A1, B1                : nat,
       Knb, Kbh    : symmetric_key,
         IMPI, IMPU,RAND,B_TID, NAF_ID, GUSS : text,
       Pkg_param, Q : nat set,
       Key_U, Key_B : message,
       X1    : {inv(public_key)}_symmetric_key

 const  sec_b_knb  : protocol_id,
       sec_b_kbh        : protocol_id,
       sec_b_ksnaf      : protocol_id,
       auth_R          : protocol_id,
       auth_B          : protocol_id

           init State := 21

     transition
```

```
            1. State  = 21 /\ RCV_BU(IMPI'.IMPU') =|>
               State':= 22 /\ SND_BH({IMPI'.IMPU'}_Kbh)
        /\ secret(Kbh,sec_b_kbh,{B,H})

            2. State  = 22 /\ RCV_BH(Kpub_u'.{X1'.RAND'.Pkg_param'
             .GUSS'}_Kbh) =|>
               State':= 23 /\ SND_BU(Kpub_u'.X1'.RAND'.Pkg_param')
         /\ secret(Kbh,sec_b_kbh,{B,H})

            3. State  = 23 /\ RCV_BU(Sign(RAND',N1',Q').N1'.Q'.Key_U')
              =|>
               State':= 24 /\ B1' :=new()
       /\ Ksnaf' := exp(B1', Key_U')
       /\ B_TID':=new()
       /\ SND_BU({B_TID'.exp(B1', Kpub_u)}_Kpub_u)
       /\ witness(B,U,auth_R,Sign(RAND',N1',Q'))
       /\ request(B,U,auth_R,Sign(RAND',N1',Q'))
       /\ secret(Ksnaf',sec_b_ksnaf,{U,B})

            4. State  = 24 /\ RCV_BN({IMPU'.NAF_ID'.Sign(B_TID',N1',Q')
             .N1'.Q'}_Knb) =|>
               State':= 25 /\ SND_BN({IMPU'.Ksnaf.GUSS.Pkg_param}_Knb)
       /\ witness(B,U,auth_B,Sign(B_TID',N1',Q'))
       /\ request(B,U,auth_B,Sign(B_TID',N1',Q'))
       /\ secret(Knb,sec_b_knb,{N,B})
       /\ secret(Ksnaf,sec_b_ksnaf,{N,B})

    end role

%%%%%%%%%%%%%%%%%%%%%% HSS %%%%%%%%%%%%%%%%%

    role hss (
U,B,H             : agent,
Kuh               : symmetric_key,     %sk
```

```
            Kpub_u           : public_key,
            SND_HB, RCV_HB   : channel(dy))
     played_by H def=

            local  State     : nat,
        Kbh              : symmetric_key,
            IMPI, IMPU, RAND, GUSS : text,
        Pkg_param, Q : nat set

            const  sec_h_kuh  : protocol_id,
                   sec_h_kbh  : protocol_id

            init State := 31

     transition

          1.    State  = 31 /\ RCV_HB({IMPI'.IMPU'}_Kbh) =|>
                 State':= 32 /\ RAND':=new()
             /\ Pkg_param':=new()
    /\ GUSS':=new()
    /\ SND_HB({Kpub_u.{inv(Kpub_u)}_Kuh.RAND'
      .Pkg_param'.GUSS'}_Kbh)
                             /\ secret(Kuh,sec_h_kuh,{U,H})
                    /\ secret(Kbh,sec_h_kbh,{B,H})

    end role

%%%%%%%%%%%%%%%%%%%% Session %%%%%%%%%%%%%%%%%%%%%%%%%%%%

    role session (
     U,N,B,H            : agent,
            Ksnaf      : symmetric_key,
Kuh                : symmetric_key,     %sk
Kpub_u             : public_key,
Sign          : hash_func,
    Request,
Initiation_Required,
Success            : text
)
    def=

      local
            SUN, SUB, SNU, SNB, SBU, SBN, SBH, SHB : channel (dy),
            RUN, RUB, RNU, RNB, RBU, RBN, RBH, RHB: channel (dy)

       composition

            ue(U,N,B,H,Ksnaf,Kuh,Kpub_u,Sign,Request,Initiation_Required,
            Success,SUN,RUN,SUB,RUB)
     /\  naf(U,N,B,Ksnaf,Request,Initiation_Required,
     Success,SNU,RNU,SNB,RNB)
            /\  bsf(U,N,B,H,Ksnaf,Kpub_u,Sign,SBU,RBU,SBN,RBN,SBH,RBH)
            /\  hss(U,B,H,Kuh,Kpub_u,SHB,RHB)

    end role

%%%%%%%%%%%%%%%%%%%%%% Environment %%%%%%%%%%%%%%%%%

role environment()
 def=

  const
           u1,n1,b1,h1            : agent,
           ksnafk,kisnafk,kuh,kih : symmetric_key,
           kpub_u                 :public_key,
           sign                   : hash_func,
           req,initiation,succ    : text

  intruder_knowledge = {u1,n1,b1,h1,kpub_u,sign,req,initiation,succ}
  composition
       session(u1,n1,b1,h1,ksnafk,kuh,kpub_u,sign,req,initiation,succ)
       /\ session(i,n1,b1,h1,kisnafk,kih,kpub_u,sign,req,initiation,succ)

    end role

    goal

     secrecy_of sec_u_kuh, sec_u_ksnaf, sec_n_knb, sec_n_ksnaf,
     sec_b_ksnaf, sec_b_knb, sec_b_kbh, sec_h_kuh, sec_h_kbh

     authentication_on auth_R
     authentication_on auth_B

    end goal

environment()
```

Appendix E

IMS-IBC with Batch Verification Scheme

In this Appendix, we present the Message Signing and Batch Verification phases of the solution proposed by Zhang et al. [Zhang 08]. Then, we present how we can use the same scheme to decrease the signature's verification time in the proposed IMS-IBC solution.

E.1 Identity-based Batch Verification (IBV)

Zhang et al. [Zhang 08] proposed a solution called Identity-based Batch Verification (IBV) scheme and it is applied to Vehicular Network. The IBV has four phases: Key Generation and Pre-distribution, Pseudo Identity Generation, Message Signing and Batch Verification. We present in this Appendix the Message Signing and Batch Verification phases.

We do not present the first two phases in this Appendix (to have more information about the scheme, please read their paper [Zhang 08]).

First of all, we present the present the parameters of their system in Table E.1.

Notation	**Description**
V^i	The ith vehicle
RSU	A roadside unit
TA	A trust authority
$\mathbb{G}$	A cyclic additive group
$\mathbb{G}_T$	A cyclic multiplicative group
P	The generator of the cyclic additive group $\mathbb{G}$
e	A bilinear map: $\mathbb{G} * \mathbb{G} \rightarrow \mathbb{G}_T$
q	The order of the group $\mathbb{G}$
r	A random nonce
s_1, s_2	The private master keys of the TA. They are stored in the vehicle's tamper-proof device
$Ppub_1$, $Ppub_2$	The public keys of the TA, $Ppub_1 = s_1.P$, $Ppub_2 = s_2.P$
$H(.)$	A MapToPoint hash [Boneh 01] function such as $H : \{0,1\}^* \rightarrow \mathbb{G}$
RID	The real Identity of the vehicle
PWD	a password used to activate the tamper proof device
ID^i	A pseudo identity of the vehicle V^i
ID^i_j	A part of ID^i since $ID^i = (ID^i_1,\, ID^i_2)$
SK^i	A private key of the vehicle V^i
SK^i_1, SK^i_2	parts of the SK^i, $SK^i_1 = s_1.ID^i_1$ and $SK^i_2 = s_2.\mathrm{H}(ID^i_1 \| ID^i_2)$
M_i	a message sent by the vehicle V^i
$h(.)$	A one-way hash function such that SHA-1
$H(.)$	A MapToPoint function, $H(.)$: $\{0,1\}^* \rightarrow \mathbb{G}$
$\|\|$	Message concatenation operation

Table E.1: System's Parameters

E.1.1 Message Signing

In the proposed IBV scheme, the message signing phase has these steps:

1) A vehicle V_i generates a message M_i.

2) V_i chooses a pseudo identity $ID^i = (ID^i_1,\, ID^i_2)$ and the tamper-proof device generates the corresponding private key $SK^i = (SK^i_1,\, SK^i_2)$.

3) V_i computes the signature σ_i of the message M_i , where

$$\sigma_i = SK_1^i + h(M_i).SK_2^i \tag{E.1}$$

4) Then, V_i sends the final message (ID^i, M_i, σ_i) to the nearest RSU.

5) These steps are iteratived every 100-300 ms.

E.1.2 Batch Verification

When an RSU receives a traffic related message from a vehicle, it has to verify the message's signature to conclude that this vehicle is not impersonating any other legitimate vehicle.

To make understanding easier, The authors first introduces the single signature verification process. Then, they presented the batch verification of multiple signatures (the latter are signed by different vehicles on various messages).

Single signature verification: The system's public parameters are $\{\mathbb{G}$, $\mathbb{G}_T$, q, P, $Ppub_1$, $Ppub_2\}$ generated by the TA and the message (ID^i, M_i, σ_i) sent by the vehicle V_i, the signature σ_i is valid if

$$e(\sigma_i, P) = e(ID_1^i, Ppub_1).e(h(M_i).H(ID_1^i||ID_2^i), Ppub_2) \tag{E.2}$$

This equation is verified because:

$$\begin{aligned} e(\sigma_i, P) &= e(SK_1^i + h(M_i).SK_2^i, P) \\ &= e(SK_1^i, P).e(h(M_i).SK_2^i, P) \\ &= e(s_1.ID_1^i, P).e(h(M_i).s_2.H(ID_1^i||ID_2^i), P) \\ &= e(ID_1^i, s_1.P).e(h(M_i).H(ID_1^i||ID_2^i), s_2.P) \\ &= e(ID_1^i, Ppub_1).e(h(M_i).H(ID_1^i||ID_2^i), Ppub_2) \end{aligned}$$

Batch verification: If we have n distinct messages defined as (ID^1, M_1, σ_1), (ID^2, M_2, σ_2), ..., (ID^n, M_n, σ_n), respectively, which are received by RSU from n distinct vehicles V_1, V_2, ..., V_n. The signatures are σ_1, σ_2, ..., σ_n. They are valid if

$$e(\sum_{i=1}^{n} \sigma^i, P) = e(\sum_{i=1}^{n} ID_1^i, Ppub_1).e(\sum_{i=1}^{n} h(M_i).H(ID_1^i||ID_2^i), Ppub_2) \tag{E.3}$$

This batch verification equation is valid since

$$\begin{aligned} e(\sum_{i=1}^{n} \sigma^i, P) &= e(\sum_{i=1}^{n}(SK_1^i + h(M_i).SK_2^i, P) \\ &= e(\sum_{i=1}^{n} SK_1^i, P).e(\sum_{i=1}^{n} h(M_i).SK_2^i, P) \\ &= e(\sum_{i=1}^{n} s_1.ID_1^i, P).e(\sum_{i=1}^{n} h(M_i).s_2.H(ID_1^i||ID_2^i), P) \\ &= e(\sum_{i=1}^{n} ID_1^i, s_1.P).e(\sum_{i=1}^{n} h(M_i).H(ID_1^i||ID_2^i), s_2.P) \\ &= e(\sum_{i=1}^{n} ID_1^i, Ppub_1).e(\sum_{i=1}^{n} h(M_i).H(ID_1^i||ID_2^i), Ppub_2) \end{aligned}$$

E.2 IMS Service Authentication Based on IBC

In this solution, we are inspired by the Identity-based Batch Verification (IBV) proposed by [Zhang 08].

We use our mechanism defined in Chapter 6, but we add the Batch Verification scheme in the Bootstrapping Server Function (BSF).

The HSS has a PKG server which has the role to generate the private keys for the UE. We use the parameters presented in Section 2.7.1 for the Bilinear pairing. We use bilinear map e: $\mathbb{G} * \mathbb{G} \to \mathbb{G}_T$. The PKG randomly generates its two master keys s_1, $s_2 \in \mathbb{Z}_q^*$, and computes its public keys $Ppub_1 = s_1.P$, $Ppub_2 = s_2.P$ which are two points $\in \mathbb{G}$.

In our work, we assume that the UE has the shared key sk with HSS and the PKG parameters (order q, prime number p, P, $Ppub_1$, $Ppub_2$ and MapToPoint function) stored in the ISIM card.

We present in Table E.2, all the notations are used in the solution (most of the mathematical notation are similar to the one presented in Table E.1).

Notation	**Description**
UE^i	The ith UE: User Equipment
$\mathbb{G}$	A cyclic additive group
$\mathbb{G}_T$	A cyclic multiplicative group
P	The generator of the cyclic additive group $\mathbb{G}$
e	A bilinear map: $\mathbb{G} * \mathbb{G} \rightarrow \mathbb{G}_T$
q	The order of the group $\mathbb{G}$
r	A random nonce
s_1, s_2	The private master keys of the PKG
$Ppub_1$, $Ppub_2$	The public keys of the PKG, $Ppub_1 = s_1.P$, $Ppub_2 = s_2.P$
$H(.)$	A MapToPoint hash [Boneh 01] function such as $H : \{0,1\}^* \rightarrow \mathbb{G}$
UEID^i	UEID^i $= H($IMPI$^i)$
$Kpub_1^i$, $Kpub_2^i$	The public keys of the UE^i
SK_1^i, SK_2^i	The private keys of UE^i, $SK_1^i = s_1.Kpub_1^i$ and $SK_2^i = s_2.Kpub_2^i$
RAND$_i$	Random value to authenticate UE^i
$H(.)$	A MapToPoint function, $H(.)$: $\{0,1\}^* \rightarrow \mathbb{G}$
$h(.)$	A one-way hash function such that SHA-1
$\|\|$	Message concatenation operation, which appends several messages together in a special format

Table E.2: Notations

The modified solution is explained in the following steps.

E.2.1 Solution's Steps

Step 1. (messages 1 and 2) UE^i starts communication with the NAF without GBA parameters. If the NAF requires the use of shared keys obtained by means of the GBA, but the request from the UE does not include GBA-related parameters, the NAF replies with a bootstrapping initiation message.

Step 2. (messages 3, 4 and 5) UE^i sends a HTTP request to the BSF (Bootstrapping Server Function) including its IMS private user identity (IMPI^i) and public user identity (IMPU^i). The BSF then retrieves from the HSS:

1. the public keys $Kpub_1^i$ and $Kpub_2^i$ (generated using IMPU^i) from the PKG.

$$Kpub_1^i = r.P \tag{E.4}$$

and

$$Kpub_2^i = UEID^i \oplus H(r.Ppub_1) \tag{E.5}$$

where r is a random number, XOR operation and UEID^i $= H$(IMPU^i)

2. the complete set of GBA user security settings (GUSS^i),
3. an Authentication Vector (AV^i) containing the RAND$_i$ and PKG parameters,
4. the private keys $Kpriv_1^i$ and $Kpriv_2^i$ encrypted with shared key sk where

$$Kpriv_1^i = s_1.Kpub_1^i \tag{E.6}$$

and

$$Kpriv_2^i = s_2.H(Kpub_1^i||Kpub_2^i) \tag{E.7}$$

Step 3. (message 6) In order to demand the UE^i to authenticate itself, the BSF forwards $Kpub_1^i$, $Kpub_2^i$, $[Kpriv_1^i]_{sk}$ and $[Kpriv_2^i]_{sk}$ and AV^i, which contains $RAND_i$ and PKG parameters, to it in the “401 (Unauthorized) message”.

Step 4. (message 7) The UE^i extracts its private keys $Kpriv_1^i$ and $Kpriv_2^i$ using the shared key sk which is stored in the ISIM card. Then, the UE^i hashes the RAND$_i$ and computes the signature Sig$_1^i$ where:

$$Sig_1^i = Kpriv_1^i + h(RAND_i).Kpriv_2^i \tag{E.8}$$

The UE^i and the BSF perform the Elliptic Curve Diffie-Hellman (ECDH) Protocol. This key agreement protocol is used to generate the Ks_NAF key. The UE^i and the BSF first have to agree whether to use the shared keys obtained by means of the GBA. The UE^i chooses a random value 'a' to generate '$a.Kpub_1^i$'. The UE^i sends IMPU^i, RAND^i, Sig$_1^i$ and '$a.Kpub_1^i$' to the BSF in an HTTP request. To verify the UE^i's signature, the BSF has already the PKG parameters and $Kpub_1^i$ and $Kpub_2^i$ corresponding to IMPU^i. Sig$_1^i$ is valid if

$$e(Sig_1^i, P) = e(Kpub_1^i, Ppub_1).e(h(RAND_i).H(Kpub_1^i||Kpub_2^i), Ppub_2) \tag{E.9}$$

If the verification phase is successful, then, the user is authenticated. This verification method is similar to the one made by Zhang et al. for equation E.2.

Step 5. (message 8) After the successful verification, the BSF generates B-TID^i (Bootstrapping ID) and stores it with the IMPU^i and GUSS^i. The BSF generates a random value 'b' and computes the value '$b.Kpub_1^i$'. Then, the BSF sends to the UE^i a "200 OK message" including the B-TID^i and '$b.Kpub_1^i$' encrypted with UE^i's public key Kpub$_1^i$ (BSF can use any asymmetric elliptic curve algorithm). After receiving the message, the UE^i retrieves the B-TID^i using $Kpriv_1^i$. Also, the UE^i and the NAF share the same Ks_NAF $= a.b.Kpub_1^i$.

In our solution, there is no key material Ks stored in the UE^i and the BSF. Our system is based on asymmetric cryptography. The shared key sk between the UE^i and the HSS is used to encrypt the UE^i's $Kpriv_1^i$ and $Kpriv_2^i$. The BSF cannot retrieve these keys and has to encrypt B-TID^i using UE^i's $Kpub_1^i$.

Step 6. (message 9) In this step, the UE^i provides the IMPU^i, B-TID^i and a signature of B-TID^i to the NAF to allow it to retrieve the corresponding keys from the BSF. The Signature of B-TID^i is:

$$Sig_2^i = Kpriv_1^i + h(B - TID^i).Kpriv_2^i \tag{E.10}$$

Step 7. (message 10) The NAF sends to the BSF the NAF-ID, the IMPU^i, B-TID^i and Sig$_2^i$ to request for GUSS^i and PKG parameters. NAF-ID is used by the BSF to verify that the NAF is authorized to use that hostname.

Step 8. (message 11) First of all, the BSF verifies the signature using $Kpub_1^i$ and $Kpub_2^i$ (same verification as in step 4, equation E.9). Then, it retrieves the GUSS^i and PKG parameters using B-TID^i and IMPU^i. Finally, it supplies to the NAF the IMPU^i, Ks_NAF, GUSS^i, and the PKG parameters.

Step 9. (message 12) The NAF checks the authentication and the authorization of the UE^i to the services according to the received GUSS^i. Once the execution of the protocol is completed, the UE^i and the NAF communicates in a secure way and UE^i is granted the services.

In Figure E.1, we illustrate all the messages exchanged to authenticate the i^{th} UE (UE^i).

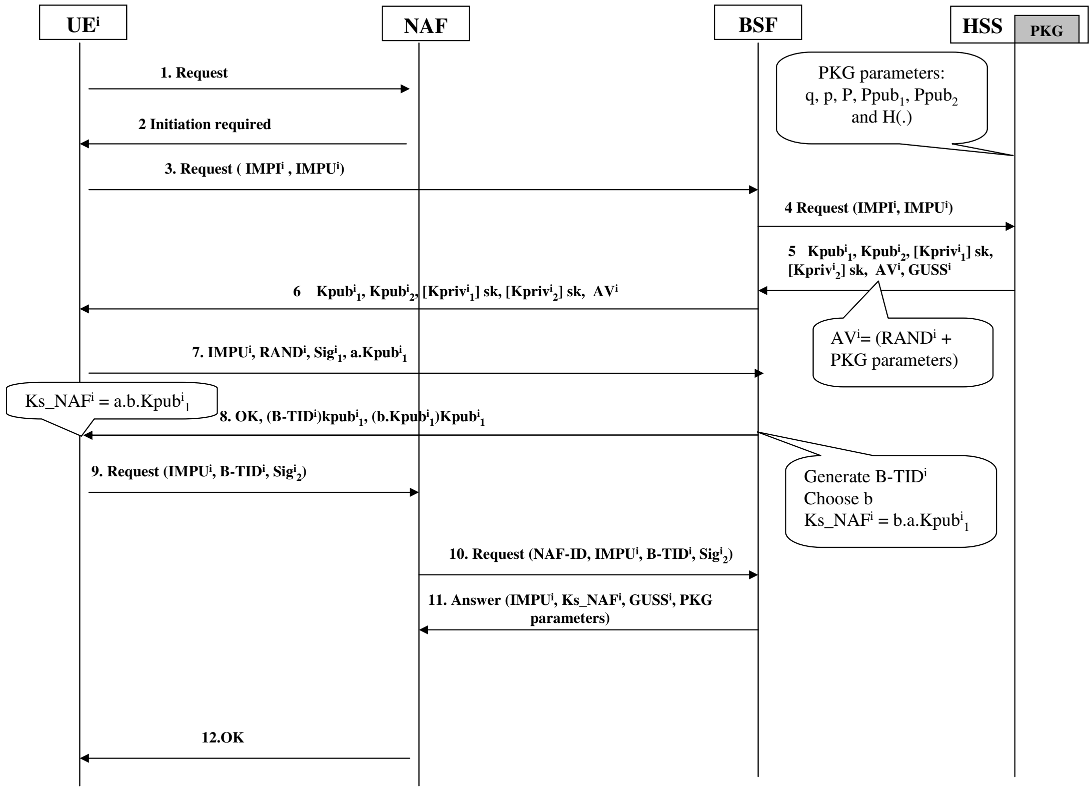

UE^i
NAF
BSF
HSS
PKG
PKG parameters: q, p, P, Ppub$_1$, Ppub$_2$ and H(.)
1. Request
2 Initiation required
3. Request (IMPI^i , IMPU^i)
4 Request (IMPI^i, IMPU^i)
5 Kpub$^i_1$, Kpub$^i_2$, [Kpriv$^i_1$] sk, [Kpriv$^i_2$] sk, AV^i, GUSS^i
6 Kpub$^i_1$, Kpub$^i_2$, [Kpriv$^i_1$] sk, [Kpriv$^i_2$] sk, AV^i
AV^i= (RAND^i + PKG parameters)
7. IMPU^i, RAND^i, Sig$^i_1$, a.Kpub$^i_1$
Ks_NAF^i = a.b.Kpub$^i_1$
8. OK, (B-TID^i)kpub$^i_1$, (b.Kpub$^i_1$)Kpub$^i_1$
Generate B-TID^i Choose b Ks_NAF^i = b.a.Kpub$^i_1$
9. Request (IMPU^i, B-TID^i, Sig$^i_2$)
10. Request (NAF-ID, IMPU^i, B-TID^i, Sig$^i_2$)
11. Answer (IMPU^i, Ks_NAF^i, GUSS^i, PKG parameters)
12.OK

Figure E.1: IMS Service Authentication for the i^{th} UE

E.2.2 Batch Verification in our Modified Solution

To verify Sig$_1^i$ and Sig$_2^i$, the BSF needs one MapToPoint hash (H), one multiplication, and three pairing operations. In [Zhang 08], they estimate that the computation cost of a pairing operation is much higher than the cost of a MapToPoint hash and a multiplication operation.

We suppose that we have n UEs which belong to the same HSS and communicate through the same BSF. The latter receives (IMPU^1, RAND$_1$, Sig$_j^1$), (IMPU^2, RAND$_2$, Sig$_j^2$), ..., (IMPU^n, RAND$_n$, Sig$_j^n$), respectively, which are sent by n distinct UEs: UE^1, UE^2, ..., UE^n and $j = 1$ or 2.

We just focus in this work on $j = 1$ because it is respectively the same for $j = 2$. Same as in [Zhang 08], all the signatures, denoted Sig$_1^1$, Sig$_1^2$, ..., Sig$_1^n$, are valid if

$$e(\sum_{i=1}^{n} Sig_1^i, P) = e(\sum_{i=1}^{n} Kpub_1^i, Ppub_1).e(\sum_{i=1}^{n} h(RAND_i).H(Kpub_1^i||Kpub_2^i), Ppub_2) \tag{E.11}$$

We verify this equation like Zhang et al. did for E.3.

Appendix F

Résumé

Dans ce chapitre, nous présentons un résumé des contributions de cette thèse.

F.1 Introduction Générale

Les mécanismes d'authentifications sont nécessaires pour sécuriser l'accès aux systèmes informatiques et aux services. L'authentification fiable des entités ou personnes clientes d'un service sert à vérifier leur identité. Les preuves de l'identité sont de trois types : ce que l'on connait (mot de passe), que l'on possède (support physique, carte à puce) et que l'on est ou que l'on sait faire (biométrie).

Nous nous intéressons, dans cette thèse, à trois problématiques applicatives émergentes : la maison intelligente (authentification des utilisateurs des équipements électroniques d'un réseau domestique) ; les documents électroniques (authentification du détenteur d'un e-passeport) ; enfin, les services multimédias (authentification des utilisateurs d'un service multimédia distant). Nous proposons, pour chacune de ces trois applications, un nouveau mécanisme d'authentification basé sur l'identité des utilisateurs. L'identité peut être des gabarits biométriques publics, des chaînes de caractères simples comme l'adresses e-mail, l'identifiant (login), etc.

La première solution concerne l'utilisation des données biométriques dans les mécanismes d'authentification dans les réseaux domestiques "Home Network (HN)". Nous voulons personnaliser l'accès de chaque utilisateur dans le HN et prévenir les utilisateurs illégitimes (en passant par la passerelle domestique "Home Gateway (HG)") d'avoir accès aux services. Nous proposons une nouvelle méthode d'authentification biométrique qui respecte la contrainte de ne pas sauvegarder les données biométriques "Biometric Template (BT)" des utilisateurs dans l'HG. Pour satisfaire cette contrainte, nous proposons d'utiliser la méthode de "Fuzzy Vault" pour cacher un secret utilisé pour l'authentification. Un logiciel génère une identité biométrique révocable (BioID) en utilisant une transformation fonctionnelle. Ce BioID est utilisée par le mécanisme du fuzzy vault pour cacher une clé de session secrète.

La deuxième solution propose des mécanismes d'authentification pour les passeports biométriques (e-Passeports). Les paramètres de chiffrement sont générés en utilisant les données biométriques et, ainsi, ils seront personnalisés pour l'utilisateur. Notre proposition introduit un nouveau mécanisme d'authentification pour le passeport biométrique utilisant le protocole Diffie-Hellman de partage de clé basé sur les courbes elliptiques (ECDH). Ce protocole est nécessaire pour générer une clé de session utilisée pour authentifier le voyageur et le Système d'Inspection (IS) et ainsi sécuriser l'échange des données entre eux. Notre protocole peut utiliser les points minuties d'une empreinte digitale et le code de l'iris du détenteur de l'e-Passport.

Dans la troisième solution, nous avons travaillé sur le réseau cellulaire et nous avons utilisé une chaîne de caractères simple (l'adresse e-mail de l'utilisateur) comme identifiant pour accéder aux services. Nous avons choisi l'IP Multimedia Subsystem (IMS) qui est une architecture de recouvrement pour la fourniture de services multimédia, comme support. Nous avons conçu un nouveau mécanisme d'authentification aux services en utilisant la cryptographie basée sur l'identité (IBC). L'objectif était d'authentifier les utilisateurs en utilisant leurs identifiants public et privé pour surmonter les faiblesses connues du protocole "Authentication and Key Agreement (AKA)". Nous nous sommes concentrés sur les tentatives d'écoute et d'usurpation d'identité qui peuvent avoir lieu dans le scénario classique de l'IMS et nous avons montré comment la solution proposée peut prévenir ces attaques. Nous avons ensuite proposé d'ajouter une vérification par lot (Batch Verification) au niveau du Bootstrapping Server Function (BSF) pour diminuer le délai de vérification des signatures et le temps de réponse de l'authentification.

F.2 Cas d'étude : La maison intelligente

Nous proposons que les équipements électroniques sensibles accessibles au sein d'un réseau domestique soient tous munis d'un capteur d'empreintes digitales et qu'un protocole soit implémenté entre ces équipements et le serveur passerelle ("home gateway") afin d'identifier et d'authentifier l'utilisateur d'un équipement. S'appuyant sur des travaux antérieurs de Ratha et al. et Uludag et al., nous proposons, de manière très judicieuse, que les identifiants biométriques soient créés par application d'une fonction de transformation/perturbation non-inversible sur l'empreinte digitale de l'utilisateur : ainsi, en cas de vol d'un identifiant, il est très simple de le révoquer et de recréer un nouvel identifiant (utilisation d'une autre transformation). L'identifiant biométrique (BioID), connu uniquement du serveur passerelle et de l'équipement client, sert alors à initier un protocole d'authentification sécurisé classique fondé sur un challenge-réponse.

Nous considérons un réseau domestique (HN) où les utilisateurs se connectent à un routeur domestique (HG) pour l'accès Internet à large bande, en utilisant n'importe quel équipement dans le HN (voir figure 4.1).

Chaque utilisateur doit enregistrer sa modalité biométrique (BT) pour être authentifié. L'objectif est de permettre à chaque utilisateur dans le HN d'avoir un accès personnalisé et l'accès à son contexte personnel. D'un point de vue opérateur, la solution proposée respecte les contraintes opérationnelles ainsi que les contraintes imposées par la CNIL (Commision Nationale de l'Informatique et des Libertés) [CNIL 07] concernant l'utilisation de la biométrie.

Supposons qu'un identificateur biométrique (BioID) est créé en utilisant le BT de l'utilisateur, et est stocké localement dans le réseau domestique (dans le HG et les équipements). Cet identifiant n'est pas transféré dans le réseau. Il convient également de remarquer que la taille de l'espace de stockage n'est pas énorme (limité au nombre de membres de la famille à la maison).

La solution proposée nécessite trois phases, qui concernent principalement la configuration des équipements, le traitement et le du BioID et finalement la phase d'authentification.

F.2.1 Phase de configuration

Dans cette phase, chaque utilisateur doit présenter sa BT (empreinte digitale) pour être manipulé et stockées dans la base de données de l'HG. L'empreinte digitale est capturée à l'aide du capteur biométrique de l'équipement de HN. Ensuite, un logiciel, génère le BioID utilisant la transformation fonctionnelle décrite dans l'article de Ratha et al.[Ratha 07]. Le BioID ne peut pas être utilisé pour retrouver l'empreinte d'origine (c'est une nouvelle disposition des points minuties). Par exemple, le BioID peut être de 384-bit (24 points de minuties). Le stockage des BioIDs des utilisateurs se fera dans un fichier spécial mis dans l'équipement et l'HG (voir figure 4.2).

F.2.2 La connexion des utilisateurs à l'HG

Chaque fois que l'utilisateur souhaite se connecter à la HG, il/elle fait une nouvelle acquisition de son empreinte digitale, afin de s'identifier, sans qu'il soit nécessaire de saisir un login ou un mot de passe. Apres l'acquisition des empreintes digitales, elles sont traitées (comme expliqué dans la phase précédente) pour générer l'identifiant de l'utilisateur BioID. L'identifiant généré est ensuite comparé à ceux stockés dans l'équipement. Si le même identifiant existe, le login correspondant est envoyé à l'HG et le processus d'authentification peut commencer (voir figure 4.3).

F.2.3 L'authentification biométrique des utilisateurs

Lorsque l'HG reçoit le login de l'utilisateur, il recherche dans sa base de données le BioID correspondant à cet utilisateur. Puis, il commence l'authentification de l'utilisateur en fonction de cet identifiant pour lui fournir un accès personnalisé. Le processus d'authentification est basé principalement sur un mécanisme Challenge-Request/ Challenge Response. Le mécanisme est une version modifiée de l'Extensible Authentication Protocol (EAP) et il se concentre uniquement sur l'échange entre l'utilisateur et l'HG. Il n'y a aucun contact avec le serveur de l'opérateur. En fait, seulement le "vault" et le défi doivent être ajoutés au message EAP-demande. Cette solution est ouverte à toute méthode EAP. Le mécanisme d'authentification est présenté dans la figure 4.4.

Les messages sont échangés comme suit :

- Un secret est généré par l'HG. Ce dernier choisit un défi pour l'utilisateur qui veut se connecter.
- La méthode du Fuzzy Vault [Uludag 06] est appliquée dans le but de construire un vault (coffre fort) pour cacher la clé secrète. Le BioID est utilisé pour créer la vault en suivant la méthode décrite dans la section 3.5.1.2.
- Le vault résultant et le défi sont transmis à l'utilisateur. L'HG ajoute un nonce pour prévenir toute attaque de répétition (replay attack).
- L'utilisateur débloque le vault en utilisant son BioID afin de récupérer la clé secrète.
- Une fois la clé secrète est trouvée, l'utilisateur transmet le défi chiffré avec la clé secrète récupérée à l'HG. Il/Elle ajoute le nonce au message.
- L'HG décrypte le défi en utilisant la clé secrète et la compare avec celle envoyé. S'ils correspondent, l'utilisateur est authentifié et il/elle reçoit un accès personnalisé.

F.3 Cas d'étude : Réseau Gouvernemental

Nous étudions, de manière approfondie, une autre utilisation possible de la biométrie: l'authentification biométrique des détenteurs d'un passeport biométrique (un e-Passeport). Dans un premier temps, une analyse synthétique des caractéristiques des passeports électroniques et des protocoles d'authentification préconisés par les organismes de normalisation ou proposés par des équipes de recherche est réalisée. L'une des limitations majeures des protocoles existants relevée est la suivante : ces protocoles (ex : le protocole OSEP), s'ils permettent de vérifier la validité d'un passeport électronique, ne garantissent pas que la personne qui présente le passeport est bien propriétaire de celui-ci.

Pour répondre à cette problématique, nous proposons deux approches utilisant des outils de cryptographie basée sur les courbes elliptiques et des informations biométriques (empreinte digitale dans un cas ; iris de l'oeil, dans l'autre). L'idée de base de ces méthodes consiste, à la génération des paramètres cryptographiques de l'e-Passeport, en utilisant les informations biométriques de l'utilisateur.

L'identification réciproque du détenteur de l'e-Passport et du Système d'Inspection est alors réalisé en remplaçant, dans le protocole OSEP, un échange de clefs basé sur le protocole Diffie-Hellman classique par un échange de clefs de type Diffie-Hellman à base de courbes elliptiques. Parallèlement, sont intégrés, au niveau du dispositif de vérification, une lecture de l'empreinte digitale de l'utilisateur, un calcul des paramètres de la courbe elliptique représentative et une comparaison de ces paramètres avec les paramètres transmis par le e Passport à l'étape précédente, afin de vérifier que le détenteur du passeport en est bien le propriétaire.

Dans le cas spécifique de l'utilisation de l'iris, nous proposons d'utiliser la méthode de Kanade et al. [Kanade 08] pour dériver les informations d'identification à partir des données d'acquisition de l'iris. La solution comprend trois phases.

La première phase est la phase d'initialisation où une courbe elliptique dans le corps galois GF(p) est générée, avec p un nombre premier. Les paramètres nécessaires pour le protocole Diffie-Hellman basé sur les courbes elliptiques (ECDH) sont enregistrés dans la puce. Cette phase est réalisée dans le bureau de l'autorité qui délivre le passeport biométrique.

La deuxième phase est l'authentification du Système d'Inspection (IS). Elle est déjà définie à la section 5.4.2.

La troisième phase est l'authentification du détenteur de l'e-Passport, où l'IS vérifie que le détenteur de l'e-Passport est authentique et non pas un fraudeur.

F.3.1 Phase d'initialisation

Dans la phase d'initialisation, l'autorité d'émission (en particulier, le Document Verifier (DV)) délivre un e-Passport pour le voyageur. La figure F.1 présente l'ensemble des entités participantes dans cette phase.

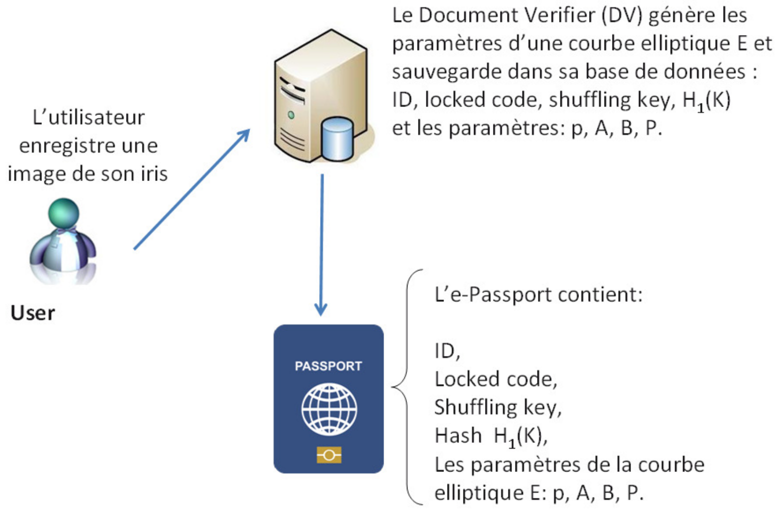

Le Document Verifier (DV) génère les paramètres d'une courbe elliptique E et sauvegarde dans sa base de données : ID, locked code, shuffling key, H_1(K) et les paramètres: p, A, B, P.
L'utilisateur enregistre une image de son iris
User
PASSPORT
L'e-Passport contient:
ID,
Locked code,
Shuffling key,
Hash H_1(K),
Les paramètres de la courbe elliptique E: p, A, B, P.

Figure F.1: Les entités participantes au protocole

Les paramètres, stockées dans la base de données utilisateur et la puce de l'e-Passport sont:

- ID: identifiant ;
- Le locked key ;
- Le shuffling key ;
- H1 (K): la valeur de hachage ;
- p : nombre premier de 128 bits (modulo) ;
- q : nombre premier de 80 bits (ordre) ;
- Les paramètres générés à l'aide de l'iris de loeil:
  - A et B: les coefficients de la courbe elliptique E;
  - P: le point public de la courbe E.

Le Document Verifier (DV) ajoute également les paramètres conventionnels pour l'e Passport comme le nom, le pays, l'âge, le sexe, ... Les paramètres A, B, p, q et P sont certifiés par le DV. Enfin, l'e-Passport est remis au voyageur. Lorsque la validité de l'e-Passport est terminée, le détenteur peut demander un nouveau et le système génère une nouvelle courbe elliptique différente de la précédente.

Le voyageur qui a besoin d'un nouvel e-Passport enrôle son iris. Pour générer les paramètres du domaine de la courbe elliptiques, le système prend en entrée les iris enrôlés. La courbe elliptique E générée ($y^2 = x^3 + Ax + B$ mod p) doit être un courbe elliptique idéale pour l'utilisation en cryptographie. Cette courbe elliptique est utilisée par la puce et le Système d'Inspection (IS) pour définir une clé de session en utilisant le protocole de partage de clé Diffie-Hellman basé sur les courbes elliptiques (ECDH).

Comme le montre la figure 3.12 (voir le chapitre 3), les paramètres de sécurité sont la clé K et la shuffling key. Celui-ci est un paramètre d'entrée dans la phase de vérification pour récupérer la valeur de la K. Le locked code est généré à partir de la clé et la clé K et le shuffling key. La procédure de génération de la courbe elliptique est présentée sur la figure F.2.

La clé K est hachée avec le Secure Hash Algorithm (SHA-256) pour créer la valeur de hachage $H_2(K)$. Cette valeur est utilisée avec un nombre premier p codé sur 128 bits et un grand nombre A codé sur 128 bits pour choisir la courbe elliptique E. approprié.

Tout d'abord, un point $P_0(X_0, Y_0)$ est généré à partir de la valeur de hachage $H_2(K)$. Comme $H_2(K)$ a 256 bits de longueur, il est coupé en deux parties, X_0 et Y_0 qui sont codées sur 128 bits. Ensuite, le DV choisit le coefficient A dans GF(p). Il définit $B = Y_0^2 - X_0^3 - aX_0$, et vérifie que $4A^3 + 27B^2$ différent de 0. Si cette condition est vérifiée, le DV calcule N = Card (E), où N est la nombre de point de la courbe. Si N est premier, un certificat de primalité est généré. Par la suite, le DV vérifie si p^j est différent de 1 mod N pour 1 <= j <= log_2p. Dans le cas négatif, le DV recommence la procédure en choisissant un nouveau coefficient A.

A la fin, le DV obtient une courbe elliptique E idéale pour les protocoles cryptographiques. Le DV choisit un point P de E qui est utilisé comme le point public de la puce. Puis, l'e-Passport est prêt à être livré au voyageur.

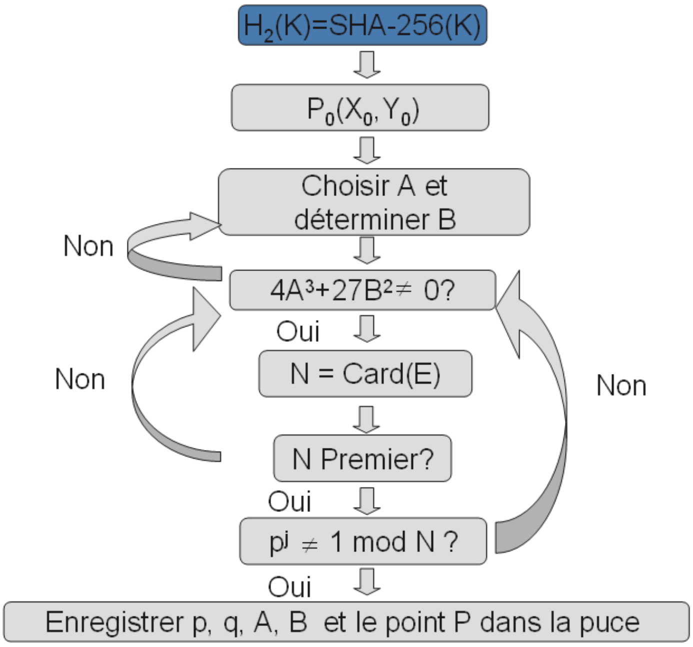


Figure F.2: Génération des paramètres de la courbe elliptique

F.3.2 Authentification du Système d'Inspection IS

L'authentification est utilisée pour vérifier l'authenticité de l'e-Passport. Cette étape est la même que l'authentification de l'IS définit dans OSEP (voir section 5.3.4.1). Nous proposons de changer le protocole de partage de clé Diffie-Hellman par le protocole de partage de clé Elliptic Curve Diffie Hellman (ECDH) Key définit dans la section 2.6.3.1. La puce C de l'e-Passport et l'IS utilisent les paramètres de la courbe elliptique pour se mettre d'accord sur une clé de session K. Puis, l'IS débute l'authentification du détenteur de l'e Passeport.

F.3.3 Authentification du détenteur de l'E-Passport

Lorsque le détenteur de l'e-Passport voyage, il lui est demandé de prouver son identité au contrôle de frontière. La procédure d'authentification est présentée dans la figure F.3.

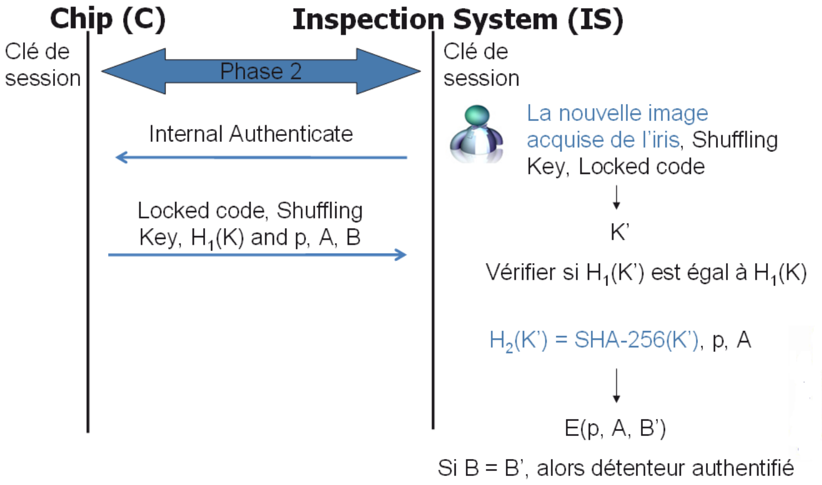

Chip (C)
Inspection System (IS)
Clé de session
Phase 2
Clé de session
Internal Authenticate
La nouvelle image acquise de l'iris, Shuffling Key, Locked code
K'
Locked code, Shuffling Key, H1(K) and p, A, B
Vérifier si H1(K') est égal à H1(K)
H2(K') = SHA-256(K'), p, A
E(p, A, B')
Si B = B', alors détenteur authentifié

Figure F.3: L'authentification du détenteur de l'e-Passport

Le voyageur fournit une donnée iris fraiche. La puce C envoie les données nécessaires au Système d'Inspection (IS) pour que ce dernier génère la courbe elliptique et ainsi authentifier le détenteur de l'e-Passport. Ces données sont, le locked code, la shuffling key, la valeur de hachage H1(K) et les paramètres de la courbe elliptique p, A et B.

Tout d'abord, l'IS génère la clé K' en utilisant la donnée iris fraichement enrôlé, la locked key et le shuffling key. L'IS vérifie si H1(K') est égal à H1(K). Ensuite, l'IS hache K 'en utilisant SHA-256 pour obtenir la valeur de hachage h2(K'). Un point P0(X0;Y0) est créé en utilisant h2(K'). Le point P0, p et A sont utilisés pour générer la courbe elliptique E'(y2 = x3 + Ax + B mod p). Si la valeur B' est égale à B, porteur de l'e-Passport est authentique. En fin de compte, l'IS et la puce C se mettent d'accord sur une clé de session extraite de K. Ainsi, la puce C peut envoyer ses données à l'IS de manière sécurisée.

Ce chapitre s'achève par une analyse de sécurité et une étude de performance (limitée à l'approche à base de données d'iris (base NIST-ICE)). Les mesures effectuées dans cette étude (temps de génération des données d'identification (3 mn). Les résultats obtenus : FRR de 0,2% (taux de faux rejets) et FAR de 3,6% (taux de fauxsses acceptations) sont satisfaisantes et l'utilisation de la biométrie de l'iris est encourageante pour le déploiement de cette solution.

F.4 Cas d'étude : Le Sous-système IP Multimédia

Dans ce cas d'étude, nous présentons un nouveau mécanisme d'authentification des services pour le "IP Multimedia Subsystems (IMS)" qui étudie l'intégration de mécanismes d'authentification et d'identification au sein de l'architecture IMS. L'objectif de cette architecture, proposée par l'Institut Européen des Normes des Télécommunications (ETSI) et le consortium de téléphonie mobile 3GPP (3rd Generation Partnership Project), est de fournir des services multimédias génériques quelle que soit la technologie de télécommunications utilisée (mobile ou fixe). Nous décrivons tout d'abord l'architecture IMS et le protocole d'authentification de services d'IMS (fondé sur la "Generic Bootstrapping Architecture (GBA)"). Puis, nous décrivons la solution où nous proposons l'intégration dans le protocole d'authentification de services d'IMS d'un mécanisme d'identification fiable utilisant la cryptographie basée sur l'identité. Ainsi, nous pouvons personnaliser le service multimédia sollicité par l'utilisateur. Nous utilisons la cryptographie basée sur les courbes elliptiques, en particulier le protocole "Elliptic Curve Diffie-Hellman". Nous profitons du fait que l'équipement utilisateur (UE) et le serveur HSS (Home Subscriber System) stockent l'ensemble des paramètres de sécurité (authentification) de l'utilisateur et partagent une clé secrète. Le protocole proposé utilise aussi la cryptographie basée sur l'identité ("Identity Based Encryption (IBC)"). Le HSS sera muni d'un générateur de clés privées ("Public Key Generator (PKG)") afin d'utiliser l'identité publique de l'utilisateur ("IP Multimedia Identity (IMPU)") pour l'identifier auprès du service cible ("Network Application Function (NAF)") et générer une clef de session personnalisée Ks_NAF. La figure 6.6 décrit le nouveau protocole nommé Authentification IMS-IBC.

Dans les messages 1 et 2, l'UE entame la communication avec le NAF sans les paramètres GBA. Si le NAF nécessite l'utilisation de clés partagées obtenus au moyen de GBA, il répond avec un message "boostrapping intitiation". Dans les messages 3, 4 et 5, l'UE envoie une requête HTTP au BSF, contenant l' "IMS private user identity (IMPI)" et l' "IMS public user identity (IMPU)". Le BSF récupère ensuite du HSS:

1. l'ensemble des paramètres de sécurité de l'utilisateur "GBA user security settings (GUSS)";
2. un vecteur d'authentification (AV) contenant une valeur aléatoire RAND et les paramètres du PKG;
3. la clé publique de l'UE Kpub = MapToPoint(IMPU) où MapToPoint est une fonction de hachage utilisée par le PKG pour convertir une chaîne de caractère en un point de la courbe elliptique E;

4. la clé privée UE Kpriv chiffrée à l'aide de la clé partagée sk. Nous notons que Kpriv = S.Kpub. avec S le secret du PKG.

Dans le message 6, le BSF transfère l'AV (RAND et les paramètres PKG), la clé publique Kpub et la clé privée Kpriv chiffrée à l'UE dans un message http “401 (unauthorized) message”. Nous avons choisi d'envoyer des paramètres du PKG à l'UE car le HSS change périodiquement ses paramètres.

Dans le message 7, l'UE récupère sa clé privée Kpriv en déchiffrant avec la clé partagée sk (stockée dans la carte ISIM). Ensuite, il génère une signature de la valeur RAND (le message m = RAND dans ce travail) à l'aide de l' “Elliptic Curve Digital Signature Algorithm (ECDSA)” (voir Section 2.6.3.3). Les entrées d'ECDSA sont:

- les paramètres du PKG (E (A, B), p, q);
- la paire de clé publique et privée de l'UE (Kpub, Kpriv);
- n: un nombre premier large qui divise le nombre de points de la courbe elliptique E;
- d = Kpriv mod (n-2);
- Q = d. Kpub

Après l'exécution de la phase de génération de signature utilisant ECDSA, la signature de la valeur RAND est la paire d'entiers (r, s). Ensuite, nous appliquons le protocole de partage de clé “Elliptic Curve Diffie-Hellman (ECDH)” (présentée dans la section 2.6.3.1). Ce protocole de partage de clé est utilisé pour générer la clé Ks_NAF. L'UE choisit une valeur aléatoire 'a' pour générer 'a.Kpub'. Il envoie Sig(RAND) = (r, s); n; Q et 'a.Kpub' au BSF dans une requête HTTP afin de s'authentifier.

Pour vérifier la signature de RAND (r, s) envoyée par l'UE, le BSF devrait suivre l'étape de vérification de signature de l'ECDSA. Si la phase de vérification est réussie, alors, l'utilisateur est authentifié. Dans le message 8, après le succès de la vérification, le BSF génère le “Bootstrapping Transaction IDentifier (B-TID)” et le stocke avec l'IMPU et le GUSS. Le BSF génère une valeur aléatoire 'b' et envoie le point 'b.Kpub' à l'UE.

Le BSF envoie ensuite à l'UE un message http “200 OK message” contenant le B-TID et la point 'b.Kpub' chiffrés avec la clé publique Kpub. L'UE déchiffre les valeurs avec la clé privée Kpriv en utilisant le protocole de Meneze-Vanstone (voir la section 2.6.3.2). Après avoir reçu le message, l'UE récupère le B-TID utilisant Kpriv et ainsi, l'UE et le BSF partage la même clé “Ks_NAF = a.b.Kpub”.

Dans le message 9, l'UE fournit l'IMPU, le B-TID, la signature de B-TID et n, Q au NAF pour lui permettre de récupérer la clé Ks_NAF, le GUSS et les paramètres du PKG du BSF.

Dans le message 10, le NAF envoie l'IMPU, le NAF-ID, le B-TID, la signature de B-TID et n, Q au BSF. NAF-ID est utilisé par le BSF pour vérifier que la NAF est autorisé à utiliser ce nom d'hôte.

Dans le message 11, le BSF vérifie la signature en utilisant Kpub. Puis, il envoie le GUSS, la clé KS_NAF, l'IMPU et les paramètres du PKG au NAF.

Dans le message 12, le NAF vérifie l'authentification et l'autorisation de l'IMPU aux services en fonction du GUSS reçu. Une fois l'exécution du protocole est achevée, l'UE et le NAF peuvent communiquer de manière sécurisée.

Une analyse de sécurité (incluant une validation du protocole à l'aide du logiciel de validation AVISPA (le code formel du protocole est disponible en annexe D)) et une étude de performance complètent ce chapitre, validant la fiabilité et la faisabilité de l'approche proposée. Une optimisation des performances est proposée via la mise en IJuvre, au niveau de la BSF, d'une procédure de vérification de lots ("batch") de signatures proposée par Zhang et al. en 2008.

Ce chapitre qui, essentiellement, propose de remplacer la phase AKA du protocole d'authentification d'IMS par une authentification à base de courbes elliptiques, démontre, chez l'auteur, des compétences certaines en analyse et ingénierie des protocoles et confirme sa bonne maîtrise des outils cryptographiques.

F.5 Conclusion

Nos travaux s'intéressent à un domaine de recherche en pleine actualité. Fondés sur l'utilisation d'outils de cryptographie et de protocoles d'authentification utilisant des courbes elliptiques et la cryptographie basée sur l'identité, ces protocoles apportent une contribution pertinente à l'état de l'art. Nous avons travaillé dans trois domaines différents: la maison intelligente, le réseau gouvernemental et les réseaux de nouvelles générations.

Pour la maison intelligente, nous avons conçu un mécanisme qui permet l'authentification des utilisateurs d'une manière distinguée ainsi que l'accès personnalisé des utilisateurs. Le mécanisme est une version modifiée de l'Extensible Authentication Protocol (EAP). Nous avons proposé de protéger les données biométriques en utilisant la biométrie cancellable. Pour partager une clé secrète, le système utilise le mécanisme de "fuzzy vault". Cette technologie devrait être utilisée avec prudence afin de protéger la vie privée des utilisateurs et empêcher la divulgation de leur donnée biométrique.

Pour le réseau gouvernemental, nous avons proposé un nouveau protocole d'authentification pour le passeport biométrique en utilisant les empreintes digitales et l'iris de l'oeil. Une chaîne de bits d'information est extraite et elle est utilisée pour générer les paramètres de sécurité du protocole cryptographique. La solution comprend trois phases où la deuxième est la même que celle définie dans la solution de Pasupathinathan et al. [Pasupathinathan 08b], sauf que nous avons utilisé le protocole d'échange de clé "Elliptic Curve Diffie-Hellman (ECDH)". Nous avons présenté une analyse de sécurité et une évaluation de la performance biométrique. Nous avons effectué des tests sur la base de données NIST-ICE d'images de l'iris pour calculer le taux de faux rejet et le taux de fausse acceptation. Les résultats obtenus (par exemple, de 0,2% FRR et FAR de 3,6%) sont satisfaisantes et l'utilisation de la biométrie de l'iris est encourageante pour le déploiement de cette solution.

Dans les réseaux de nouvelles générations NGN, nous avons utilisé la cryptographie basée sur l'identité pour améliorer le protocole d'authentification aux services pour le IP Multimedia Subsystem (IMS). Cette identité est une chaîne simple (exp: adresse e-mail ,...). L'IMS est une architecture qui fournit des services multimédias (tels que la Voix sur IP (VoIP), la vidéoconférence, présence, push-to-talk, etc.) au-dessus de tous les réseaux IP et les réseaux NGN. La sécurité est assurée grâce à un protocole symétrique avec une clé partagée (ks) entre l'User Equipement (UE) et le Home Subscriber Server (HSS), et le protocole de partage de clé Elliptic Curve Diffie-Hellman. Nous avons étudiés les tentatives d'écoute et d'usurpation d'identité et nous avons montré comment la solution proposée peut empêcher ces attaques. Nous avons ensuite proposé d'ajouter une vérification de signature par lot (batch verification) au niveau du Bootsrapping Server Function (BSF) pour diminuer le délai de vérification de signature et le temps de réponse de l'authentification.

Comme travaux futures, il y a des pistes de recherches envisageables comme :

1. Un contrôle d'accès renforcé basé sur la vie privée qui devrait être étudié et plus étendu pour supporter les préférences de confidentialité.
2. L'utilisation de la multibiométrie dans les passeports biométrique qui contiennent déjà différents types de données biométriques telles que les empreintes digitales, visage, iris, etc
3. Une contribution dans le secteur des visas électroniques (e-Visa)

Appendix G

Publications

Journals:

- Abid, M., Afifi, H, '*Towards a secure E-passport protocol based on biometrics*', Journal of Information Assurance and Security JIAS, Volume 4, Issue 4 (Special Issue on Access Control and Protocols), pages: 338-345, June 2009.
- Abid, M., Song, S., Moustafa, H. and Afifi, H., '*Integrating Identity-Based Cryptography in IMS Service Authentication*', International Journal of Network Security & Its Applications (IJNSA), Volume 1. Number 3, pages: 1- 13, October 2009.

Conference Papers:

- Abid, M., Moustafa, H., Afifi, H., Bourdon, G., '*Fuzzy Biometric Authentication in Home Networks for Personalized Users' Access*', in IEEE (ed.), 15th International Conference on Telecommunications ICT 2008, (Saint Petersburg, June 2008).
- Abid, M., Afifi, H., '*Secure E-passport Protocol using Elliptic Curve Diffie-Hellman Key Agreement Protocol*', in IEEE (ed.), IAS 2008: The Fourth International Conference on Information Assurance and Security, pages:99-102, Naples, Italie, September 2008.
- Abid, M., Song, S., Moustafa, H. and Afifi, H., '*Efficient Identity-Based Authentication for IMS based Services Access*', the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM2009), pages: 278-284, Kuala Lumpur, Malaysia, December 14-16, 2009.
- Abid, M., Kanade, S., Petrovska-Delacrétaz, D., Dorizzi, B., and Afifi, H. '*Iris Based Authentication Mechanism for e-Passports*', the 2nd International Workshop on Security and Communication Networks (IWSCN 2010), pages: 77-81, Karlstad, Sweden, May 26?28, 2010.

Research reports:

- Abid, M., Afifi, H., '*Etat de l'art sur les systèmes Crypto Biométriques*', Rapport de recherche N 07005 RS2M, May 2007
- Yang, D., Abid, M., Song, S., '*Implementation of IMS-IBC Service Authentication Platform*', Rapport de recherche N 10003 RS2M, February 2010.

Poster:

- Abid, M., Afifi, H., '*Un nouveau protocole sécurisé pour les passeports biométriques*', Workshop Interdisciplinaire sur la Sécurité Globale WISG'10, Troyes, 26- 27 janvier 2010.