A SURVEY ON CRYPTOGRAPHIC PROTOCOLS USING PAIRING-BASED CRYPTOGRAPHY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ŞEYMA FETVACI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

SEPTEMBER 2019

Approval of the thesis:

A SURVEY ON CRYPTOGRAPHIC PROTOCOLS USING PAIRING-BASED CRYPTOGRAPHY

submitted by **ŞEYMA FETVACI** in partial fulfillment of the requirements for the degree of **Master of Science in Cryptography Department, Middle East Technical University** by,

Prof. Dr. Ömür Uğur _____
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Ferruh Özbudak _____
Head of Department, **Cryptography**

Assoc. Prof. Dr. Murat Cenk _____
Supervisor, **Cryptography, METU**

Examining Committee Members:

Assoc. Prof. Dr. Ali Doğanaksoy
Mathematics, METU _____

Assoc. Prof. Dr. Murat Cenk
Cryptography, METU _____

Assist. Prof. Dr. Eda Tekin
Business Administration, Karabük University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ŞEYMA FETVACI

Signature :

ABSTRACT

A SURVEY ON CRYPTOGRAPHIC PROTOCOLS USING PAIRING-BASED CRYPTOGRAPHY

Fetvacı, Şeyma
M.S., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Murat Cenk

September 2019, 30 pages

With the thousands of works on pairing-based cryptography, the purpose of using pairings in the protocols/schemes have changed. Before, they were used just to attack the systems. Nowadays, they have been used to design such new cryptosystems that there were no applicable methods before for these protocols like Joux's key agreement scheme. The main purpose of this thesis is to analyze how some of these protocols use pairings-based cryptography in their schemes and what they achieve with these schemes. We further share some notes that should be borne in mind while establishing new protocols.

Keywords: pairing-based cryptography, bilinear map, identity-based encryption, key agreement

ÖZ

EŞLEŞTİRME TABANLI ŞİFRELEME KULLANAN KRİPTOGRAFİK PROTOKOLLER ÜZERİNE BİR ARAŞTIRMA

Fetvacı, Şeyma
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Murat Cenk

Eylül 2019, 30 sayfa

Eşleme tabanlı şifreleme üzerine yapılan binlerce çalışma sonucunda, protokollerde eşlemenin kullanılma amacı değişmiştir. Önceden sadece sistemlere saldırmak için kullanılan eşlemeler; günümüzde ise, yeni şifreleme sistemleri tasarlarken kullanılıyolar. Hatta Joux anahtar anlaşma şeması gibi protokollerin bazılarını uygulamak önceden imkansızdı. Bu tezin temel amacı, bu protokollerin bazılarının eşleşmelere dayalı kriptografiyi programlarında nasıl kullandıklarını ve bu programlarla neler başardıklarını analiz etmektir. Güvenlik açısından yeni protokoller oluştururken akılda tutulması gereken bazı notları daha da paylaşıyoruz.

Anahtar Kelimeler: eşleme tabanlı kriptografi, doğrusal gönderim, kimlik-bazlı şifreleme, anahtar anlaşması

To strong women who face every challenge in the name of all women and children

ACKNOWLEDGMENTS

I would like to express my special thanks to my supervisor Assoc. Prof. Dr. Murat Cenk for his motivation and encouragement he has given me during my thesis study.

I also would like to thank to Buse Taşcı for accompanying me almost every day during writing my thesis process, her support and help.

Beyond all, I can't thank my family and my dearest friends, especially Duygu, Gerçem, Seden and Oğuz enough. There were so many times that I had struggled; however, it was their endless support, care, patience and trust that got me back on my feet and showed my way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DLP	Discrete Logarithm Problem
DHP	Diffie-Hellman Problem
DDHP	Decisional Diffie-Hellman Problem
CDHP	Computational Diffie-Hellman Problem
GDHP	Gap Diffie-Hellman Problem
BDHP	Bilinear Diffie-Hellman Problem
BCDHP	Bilinear Computational Diffie-Hellman Problem
BDDHP	Bilinear Decisional Diffie-Hellman Problem
BDHDHP	Bilinear Decisional Hash Diffie-Hellman Problem
PBC	Pairing-Based Crytography
IBE	Identity-Based Encryption
TTP	Trusted Third Party
CCA	Chosen Ciphertext Attack
PEKS	Public-Key Encryption with Keyword Search
BLS	Boneh, Lynn and Shacham
MOV	Menezes, Okamoto and Vanstone
FR	Frey and Ruch

CHAPTER 1

INTRODUCTION

Identity-Based cryptosystems allow any couple of participants to interact safely with each other and check each other's signatures without exchanging public or private keys, without maintaining any key directories or using any third party services. In this system, with no need of any certificate, a person's identity can be used as public key such as an email address. Hence, if one user needs to comminicate with someone, s/he can use the email address of the other person directly since the public keys are decided in advance with the unique information.

After the first ID-Based cryptosystem of Shamir and some applications like identity-based encryption, searchable encryption, attribute-based encryption, short signatures were proposed, the interest in pairings in cryptography has been sharply increasing. With the thousands of works on this area, pairings where any pair of points on elliptic curves are mapped into finite fields were started to be used not just to attack the cryptosystems but also to design whole new protocols and schemes that were infeasible to apply before, such as IBE [5] and Joux's key agreement scheme [16]. With the use of pairings, finite fields are now large enough to make the cryptographic hard problems difficult to be computed as well as they are small enough to make the computations efficient on them.

In this analysis, we have investigated some of the cryptographic protocols that benefit from the pairings. Before that, to understand the process of the protocols, you will be given some mathematical backgrounds in Chapter 2. First, you will be provided the basics of elliptic curve over finite fields. Then, we will state what the bilinear map is, its types and properties. Later, we will show some security related hard

problems. Since this analysis is about pairings, the basic cryptographic concepts such as public key encryption, key exchange protocols and digital signatures are assumed to be familiar. For further information about these concepts, you may read [17].

Weil and Tate Pairing are the most used two pairings. After giving the basics of these essentials pairings in Chapter 2, we will concentrate on some cryptographic schemes that make use of these pairings in Chapter 3. This chapter will include three main sections: Encryption, Signature, Key Agreement. We will give two examples per each section. In the encryption section, we will talk about Identity-Based Encryption(IBE) and Public Key Encryption with Keyword Search(PEKS). IBE allows the sender to make sure that s/he uses the authentic copy of the receiver's private key like an email address. PEKS allow the user to search for encypted keyword and while doing the search, it doesn't jeopardize the safety of the original data. And then, in the signature section, we will move on to the Boneh, Lynn and Shacham's(BLS) Short Signature Scheme. As it is understood from the name, by using BLS signature scheme, the receiver can certify the authenticity. This scheme is also used for the aggragetion of signatures which will be also mentioned. Finally, in the last section, key agreement, we will focus on Joux's One Round 3-Party Key Agreement and Extending Joux's Protocol to Multi Party Key Agreement.

Among all these protocols, no matter what they are used for, the security is one of the important challenges. In pairing-based cryptosystems, the security is based on how much difficult to compute various computationally hard problems associated with a particular one: Discrete Logarithm Problem which they will be stated in Chapter 2. Like a butterfly effect, every choice made during the building process of protocols has an impact on the security level. To reach the desired security level, for each possible pairing, elliptic curve, embedding degree, the complexity of the operations over the finite field, and so on, the security level should be found and stated. Later, the protocols should be continued with the best choice among them. In literature, there are already some commonly wrong use of pairings that may end up with security breaches. You will see some notes why one should be careful about these choices while setting pairings in Chapter 4.

In this thesis, our aim is to analyze how protocols use pairing-based cryptography in

their schemes and what they achieve with these schemes. In this respect, we examine some of the most valuable schemes regarding encryption, signature and key agreement. We further share some notes that should be borne in mind while establishing new protocols for the sake of security.

CHAPTER 2

PRELIMINARIES

You will be first given some mathematical backgrounds related to the protocols using pairings. We will mostly concentrate on Bilinear Maps, in other words, pairings, since its properties provide a great easiness in protocols. Then, we will continue with how pairings is(or should be) used in cryptography and what hard problems it depends on for the security.

2.1 Elliptic Curves over Finite Fields

For cryptographic use, while defining the elliptic curves, they are considered over finite fields instead of real numbers. Now, we will see the basic construction of the elliptic curves over finite fields[15].

Definition 2.1. An elliptic curve over $\mathbb{Z}_p$, where $p > 3$, is the pair set $(x, y) \in \mathbb{Z}_p$ which appeases the Weierstrass equation

$$y^2 \equiv x^3 + ax + b \mod p \tag{2.1}$$

along with an imaginary point of infinity $\mathcal{O}$ where $a, b \in \mathbb{Z}_p$ and $4a^3 + 27b^2 \neq 0 \mod p$ holds.

Group Law for an ellpitic curve E over F

- *Identity:* $Q + \infty = \infty + Q = Q$ for all $Q \in E(F)$
- *Negatives:* If $Q = (x, y) \in E(F)$, then $Q + (-Q) = \infty$ where $-Q$ denoting $(x, -y)$ is another point in $E(F)$ and the negative of Q.

- *Point Addition:* Assume $Q = (x_1, y_1) \in E(F)$, $P = (x_2, y_2) \in E(F)$ and $Q \neq \pm P$. Then, $Q + P = (x_3, y_3)$ where

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \text{ and } y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$$

- *Point Doubling:* Assume $Q = (x_1, y_1) \in E(F)$ and $Q \neq -Q$. Then, $2Q = (x_3, y_3)$ where

$$x_3 = \left(\frac{3{x_1}^2 + a}{2y_1}\right)^2 - 2x_1 \text{ and } y_3 = \left(\frac{3{x_1}^2 + a}{2y_1}\right)(x_1 - x_3) - y_1$$

2.2 Bilinear Map

Definition 2.2. Let G_1 and G_2 be two additive groups, and G_T be a multiplicative group of prime order q. Assuming P_1 is a generator of G_1 and P_2 is a generator of G_2, we consider $\hat{e}$ is a *bilinear map* or *pairing* as follows:

$$\hat{e} : G_1 \times G_2 \to G_T \tag{2.2}$$

where the bilinear maps have three useful properties:

1. *Bilinearity:* $\forall P_1 \in G_1 \text{ and } P_2 \in G_2, \ \forall a, b \in \mathbb{Z}_q^*,$

$$\hat{e}(aP_1, bP_2) = \hat{e}(P_1, P_2)^{ab}$$

2. *Non-degeneracy:* We need to make sure that nothing maps to identity:

$$\forall P_1 \in G_1, \ \forall P_2 \in G_2, \ P_1 \neq 0 \text{ and } P_2 \neq 0,$$
$$< \hat{e}(P_1, P_2) > \ = \ G_T \text{ meaning that } \hat{e}(P_1, P_2) \text{ generates } G_T$$

   In other words:

$$P_1 \neq 0 \text{ and } P_2 \neq 0 \Rightarrow \hat{e}(P_1, P_2) \neq 1$$

3. *Computability:* $\hat{e}$ can be executable efficiently.

Weil and Tate pairings are two of pairings where these properties hold for an elliptic curve group G_1 and a finite field G_2. [20].

Considering the design of pairing-based protocols with respect to the particular requirements, there are mainly 4 types of bilinear maps. [18]

- Type-1: In this case, $G_1 = G_2$ and for the elements of G_1, there are no short representations.
- Type-2: In this case, $G_1 \neq G_2$ and $\phi : G_1 \rightarrow G_2$ is an efficiently computable homomorphism where there is no way of efficient secure hashing to the elements of G_2.
- Type-3: For this case, $G_1 \neq G_2$ and there is no efficiently computable homomorphism $\phi : G_2 \rightarrow G_1$
- Type-4: In this case, $G_1 \neq G_2$ and an homomorphism $\phi : G_1 \rightarrow G_2$ can be efficiently computable just like in Type-2; however, in this case, there exists an efficient secure hashing to to a group element. Because of its inefficiency, Type-4 is not usually used in protocols.

Type	Hash to G_2	Short G_1	Homomorphism	Poly time generation
1 (small char)	✓	×	✓	×
1 (large char)	✓	×	✓	✓
2	×	✓	✓	✓
3	✓	✓	×	✓

Figure 2.1: Properties of types of bilinear maps [13]

The pairings where $G_1 = G_2$ are said to be *symmetric pairings*. If not, they are called *assymmetric pairings*.

2.3 Security-Related Problems

In cryptography, there are some hard problems that are used for the security purpose while designing the protocols. The harder to solve these problems, the harder to find a breach to attack the protocols. Now, you will be provided some definitions as an instance of these problems [10].

Definition 2.3. Assume that G is a group. Given $h \in G$ such that $g^x = h$, finding x which is a hard problem is known as Discrete Logarithm problem (DLP). In other words, given P_1 and P_2, where P_1 is the generator of the group G, such that $P_2 = xP_1$, finding x is a DLP.

The difficulty of some computationally hard problems related to the DLP provides the security of certain utilizations of bilinear pairings[18].

You can find some examples of these hard problems below. For the definitions, assume that $\hat{e}$ is a bilinear pairing on $(G_1 \times G_2, G_T)$.

Definition 2.4. Given the element P_1 and the values kP_1, lP_1 for some $k, l \in \mathbb{Z}_q^*$, computing $\hat{e}(P_1, P_1)^{kl}$ is a bilinear Diffie-Hellman problem (BDHP).

Definition 2.5. Given the values P_1, kP_1, lP_1 and mP_1 for some $k, l, m \in \mathbb{Z}_q^*$, computing $\hat{e}(P_1, P_1)^{klm}$ is a bilinear Computational Diffie-Hellman problem (BCDHP).

Definition 2.6. Given the values P_1, kP_1, lP_1 and mP_1, determining if $\hat{e}(P_1, P_1)^m = \hat{e}(P_1, P_1)^{kl}$ is a bilinear Decisional Diffie-Hellman problem (BDDHP).

Definition 2.7. Given the values P_1, kP_1, lP_1, mP_1 and r for some $k, l, m, r \in \mathbb{Z}_q^*$ and a one way hash function $H : G_2 \rightarrow \mathbb{Z}_q^*$, determining if $r = H(\hat{e}(P_1, P_1)^{klm}) mod q$ is a bilinear Decisional Hash Diffie-Hellman problem (BDHDHP).

We will also talk about how these problems affects the security of the protocols in Chapter 4.

2.4 Using of Pairings in Cryptography

Earlier, pairings were used to reduce DLP over an elliptic curve to DLP over a finite field. In this way, one can attack the problems as a subexponential index calculus attack. However, with the advance works on pairings, now, they are used to design protocols by using hard problems and choosing the points where finite fields are sufficiently large enough.

2.5 Pairings Used In Cryptography

The most known and used pairings in cryptography are Weil and Tate pairings. In this analaysis, we will just give the main idea behind them to make the connections among the protocols more smooth with these pairings.

2.5.1 The Weil Pairing

Weil Pairing is the first one who described the first pairing on elliptic curves. Although it is not directly used in the cryptography, the most used pairings, Tate pairing or its variants, uses the weil pairing [11]. Here is the basis of the Weil pairing.

Theorem 2.1. *Consider E as an elliptic curve defined over a finite field K, $r \geq 2$ an integer prime to the characteristic of K, and P and Q two points of r-torsion on E. Then,*

$$\hat{e}_{W,r} = (-1)^r * \frac{f_{r,P}(Q)}{f_{r,P}(P)} \tag{2.3}$$

is well defined when $P \neq Q$ and $P, Q \neq 0_E$. One can extend the application to the domain $E[r] \times E[r]$ by requring that $\hat{e}_{W,r}(P, 0_E) = \hat{e}_{W,r}(0_E, P) = \hat{e}_{W,r}(P, P) = 1$. Furthermore, the application $\hat{e}_{W,r} : E[r] \times E[r] \mapsto \mu_r$ obtained in this way is a pairing, called the Weil Pairing. The pairing $\hat{e}_{W,r}$ is alternative, which means that $\hat{e}_{W,r}(P, Q) = \hat{e}_{W,r}(Q, P)^{-1}$.

For proof, see [[22], Section III.8] or Section 3.4.3.

Please not that the Weil pairing is defined over any field K where r is its characteristic prime and the values are in $\mu_r \subset \bar{K}$. However, it is considered as $K = \mathbb{F}_q$ with a prime number q and embedding degree k such that the smallest field containing μ_r is $\mathbb{F}_q^k$.

2.5.2 The Tate Pairing

Tate defined the Tate pairing in [23]. Assume $K = \mathbb{F}_q$, like in the Weil pairing, with a prime q and the embedding degree k corresponding to r.

Theorem 2.2. *Consider E as an elliptic curve where r is a prime number dividing the number of $E(\mathbb{F}_q)$, and $P \in E[r](\mathbb{F}_q^k)$ is a point of r-torsion defined over $\mathbb{F}_q^k$ and $Q \in E(\mathbb{F}_q^k)$ is a point of elliptic curve described over $\mathbb{F}_q^k$. And let R be any point in $E(\mathbb{F}_q^k)$ where $\{R, Q+R\} \cap \{P, 0_E\} = 0$. Then,*

$$\hat{e}_{T,r}(P,Q) = \frac{f_{r,P}(Q+R)}{f_{r,P}(R)}^{\frac{q^k-1}{r}} \tag{2.4}$$

is well defined and independent to R.

Moreover, the application

$$\begin{aligned} E[r](\mathbb{F}_q^k) \times E(\mathbb{F}_q^k)/rE(\mathbb{F}_q^k) &\to \mu_r \\ (P,Q) &\mapsto \hat{e}_{T,r}(P,Q). \end{aligned} \tag{2.5}$$

is a pairing, called the Tate Pairing.

For the proof the theorem, please see [12]. And also please see [11] to understand the usage of Weil and Tate Pairings in cryptography for the applications of the protocols.

CHAPTER 3

CRYPTOGRAPHIC PROTOCOLS USING PAIRING-BASED CRYPTOGRAPHY

In this chapter, we concentrate on some of the significant cryptographic applications depending on two known bilinear pairings, Weil and Tate Pairings, in three different parts. In the first part, we describe two of the important encryption schemes, namely *ID-based Encryption* and *Public Key Encryption with keyword Search*. In the second part, we continue with one of the significant signature schemes *Boneh, Lynn and Shacham Short Signature Scheme* and how it is used for the aggregation of signatures. In the last part, we first focus on the key agreement protocols *Joux's One Round 3-Party Key Agreement Scheme* and then focus on a related one, namely *Extending Joux's Protocol to Multi Party Key Agreement*.

3.1 Part I: Encryption

The secrecy of a message has been important to us since the beginning of the life. From the individuals to the companies or governments during the history, there are so many things that are needed to be kept secret. To be able to this, we need to encrypt these messages. In this part, we will see two encryption schemes that use pairing based cryptography while encryting the messages.

3.1.1 Identity-Based Encryption

When Bob wants to send a secured message to Alice by using public-key encryption, he does the process using Alice's public key. Then, when Alice wants to decrypt the message, she uses her corresponding private key. In this process, Bob has to be sure of using authentic copy of Alice's private key. Otherwise, a hostile could get in their way, persuade Bob to use attacker's public key and thereby the attacker could decrypt Bob's message that was meant to be sent to Alice. [20]

To decrease the inherent problems of managing certificates, in 1984, Adi Shamir proposed that the public keys of the recepients should contain the identifying information, such as an email address. Now, Bob could send a secure message to Alice without the prior setup of a trusted third party, TTP. He even sends the message even before Alice generates a key pair.

The first practical IBE scheme was offered by Boneh and Franklin in 2001. The scheme utilizes a bilinear pairing $\hat{e}$ on (G_1, G_T) by using symmetric pairings. [5]

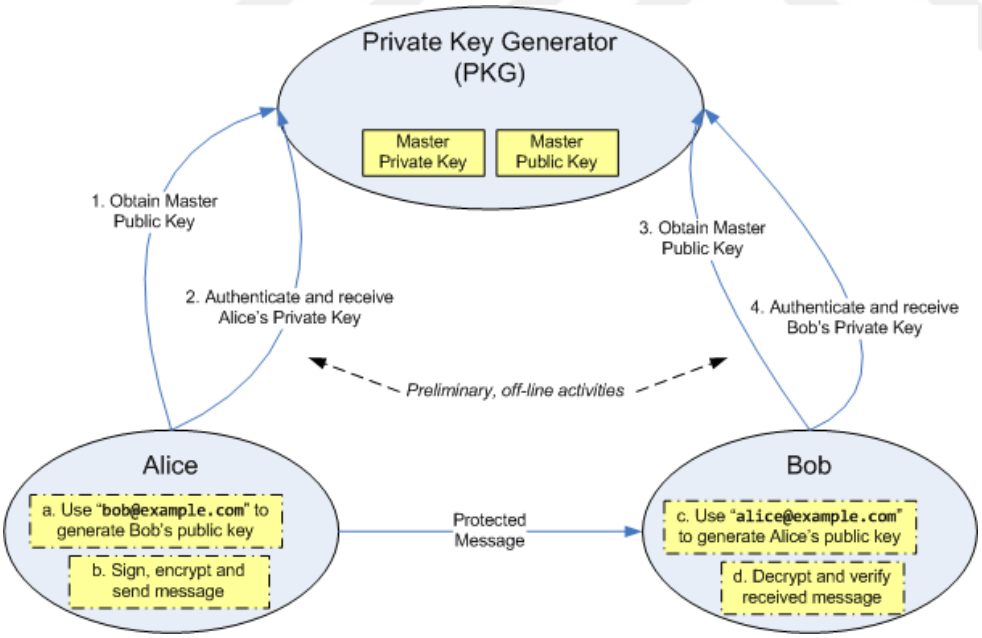

Figure 3.1: How IBE works. [5]

Setup, Extract, Encrypt and Decrypt are the 4 random algorithms that specify the IBE scheme. Now, let's look at them shortly.

Setup

- uses the symmetric bilinear mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$ and P as a generator.
- chooses a secret key s system-wide.
- sets a corresponding system-wide public key $P_{pub} = sP$.

Extract

- takes the parameters from Setup, chooses a public identity $A \in \{0,1\}^*$ and outputs a private key d_A. By using this algorithm, a private key is extracted from the provided public key.

Encrypt

- By using the parameters above, the aim here is to encrpyt a message m to public key A.

$$Enc(P_{pub}, A, m) = \langle rP, m \oplus H_2(g_A^r) \rangle, r \in_R \mathbb{Z}_q^*$$
$$g_A = \hat{e}(Q_A, P_{pub})$$
$$Q_A = H_1(A)$$
$$H_1 : \{0,1\}^* \mapsto G_1, \text{ a random oracle}$$
$$H_2 : G_2 \mapsto \{0,1\}^*, \text{ a random oracle}$$

Decrypt

- The aim of this algoritm is to decrypt $c = (u, v)$. The secret key mentioned in Setup is given to the holder of A as $d_A = sQ_A$ where $Q_A = H_1(A)$. Then,

$$\begin{aligned} Dec(u, v, d_A) &= v \oplus H_2(\hat{e}(d_A, u)) \\ &= v \oplus H_2(\hat{e}(sH_1(A), rP)) \\ &= v \oplus H_2(\hat{e}(H_1(A), P)^{rs}) \\ &= v \oplus H_2(\hat{e}(Q_A, sP)^r) \\ &= v \oplus H_2(\hat{e}(Q_A, P_{pub})^r) \\ &= v \oplus H_2(g_A^r) \\ &= (m \oplus H_2(g_A^r)) \oplus H_2(g_A^r) \\ &= m \end{aligned} \tag{3.1}$$

The security of this scheme is based on the difficulty of BDHP.

Ran Canetti and Ron Rivest [8] assumed that this scheme can be made CCA2_secure with Fujisaki-Okamoto construction.

CCA2_secure means that the algorithm is secure against an adaptive chosen ciphertext attack where the attackers can make their choices of the plaintexts to the decryption algorithm that relies on the earlier chosen ciphertext queries. [2]

3.1.2 Public Key Encryption with keyword Search

Assume that the user Alice wants to read her emails from different devices such as laptop, desktop, pager, etc. Alice's mail gateway is expected to direct an email to the corresponding machine depending on the email keywords. For example, when an email containing keyword "crucial" is sent to Alice by Bob, the email should be directed to her pager. Or when an email containing keyword "dinner" is sent to her, the email should be directed to her desktop so that she can read it later. So, the expected thing here is that each email should include a few of keywords such as words on subject lines or even the email address of the sender.

Let's say Bob sends an ciphered email to Alice by using her public key where the email and the related keywords are encrypted. This makes it impossible for the email gateway to see the keywords and direct them to the correct devices. The goal in this protocol is to give Alice an option where she can give the gateway the ability to test if "crucial" is included in the email as a keyword; however, while testing this, the gateway cannot learn anything else from the encrypted email. Now, we will see how this works.

First, Bob uses a standard public key system, encrypts his message and appends a *Public-Key Encryption with keyword Search* (PEKS) per keyword to the ciphertext [4]. In other words, for message M and keywords $K_1, ..., K_m$, Bob sends

$$E_{A_{pub}}(M) \quad || \quad PEKS(A_{pub}K_1) \quad || \quad ... \quad || \quad PEKS(A_{pub}, K_m)$$

where Alice's public key is denoted by A_{pub} and $E_{A_{pub}}(M)$ is the ciphertext. This way, PEKS enables Alice to give the gateway a cretain trapdoor T_K. Given $PEKS(A_{pub}, K')$ and $E_{A_{pub}}(M)$, the gateway can test if $K = K'$. As noticed, there is no communication between Bob and Alice in this process.

The scheme in [4] consists of 4 algorithms, namely: KeyGen, PEKS, Trapdoor and Test. And it is built as a non-interactive searchable encryption. They use $\hat{e} : G_1 \times G_1 \rightarrow G_2$ which is a bilinear map and $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : G_2 \rightarrow \{0,1\}^{logp}$ which are hash functions.

KeyGen

- The given security variable decides the size of G_1 and G_2, namely p. After a random $\delta \in \mathbb{Z}_p^*$ and a generator g of G_1 are picked by the algorithm, it outputs $A_{pub} = [g, h = g^{\delta}]$ and $A_{priv} = \delta$.

PEKS(A_{pub}, K)

- The algorithm calculates $t = \hat{e}(H_1(K), h^r) \in G_2$ for a random $r \in \mathbb{Z}_p^*$ where PEKS$(A_{pub}, K) = [g^r, H_2(t)]$ is the output.

Trapdoor(A_{priv}, K)

- outputs $T_K = H_1(K)^\delta \in G_1$.

Test$((A_{pub}, S, T_K)$

- Let $S = [A, B]$. Check if $H_2(\hat{e}(T_K, A)) = B$. If it holds, outputs "YES"; otherwise, outputs "NO".

This system, in the random oracle model, is stated semantically secure against a chosen keyword attack. And it is also a non-interactive searchable encryption scheme. The BDHP that we mentioned in Chapter 1 provides the security of this system.

Boneh et al give another construction on this encryption technique, which is less efficient, called a limited system based on general trapdoor permutation [4], which will not be mentioned on this survey.

3.2 Part II: Signature

When a person is requested a handly-operated key in the signature, keeping the signature short is desirable. Or in schemes with restricted bandwidth, short digital signatures are preferred. RSA and DSA are the two most frequently used signature schemes. If users use 1024-bit modulus, the length is 1024 bits long for RSA signature while the length of DSA signatures is 320 bits long. Still, they are too long to enter the key manually. The BLS scheme with a signature length of about 160 bits achieves approximately the same level of security as in the DSA scheme. [7]

3.2.1 Boneh, Lynn and Shacham (BLS) Short Signature Scheme

The BLS scheme utilizes bilinear mapping for the verification process and operates in any group where Decisional DiffiHellman Problem is easy but Computational Diffie-Hellman is hard. These groups are called "Gap Groups". Thereby, it is provably secure.

The scheme consists of 3 functions, Key Generation, Signing, Verification, and uses $H_1 : \{0,1\}^* \mapsto G_1$ which is a full-domain hash function and it is considered as a random oracle. Now, let's see the basic idea behind the short signatures. [6]

Key Generation

- Under a Co-GDH setup, picks a random integer $x \xleftarrow{R} \mathbb{Z}_q$ as a secret key, and calculates $v = g^x$ and takes it as public key.

Signing

- Given x, first calculates $h = H(m)$ where $h \in G_1$ and a message $m \in \{0,1\}$, and then the signature $\sigma = h^x$.

Verification

- Given v, the signature σ, calculates $h = H(m)$ where m is the message, and justifies that $\hat{e}(\sigma, g) = \hat{e}(h, v)$.

The BLS scheme also makes the aggregation of signatures available. Here is the scheme for the Bilinear Aggregate Signatures [6]:

Key Generation

- Under the help of Co-GDH setup, for a specific participant, takes a random integer
  $x \xleftarrow{R} \mathbb{Z}_q$ as a secret key, and calculates $v = g^x$ and takes it as public key.

Signing

- For a particular user, given x, first calculates $h = H(m)$ where $h \in G_1$ and a message $m \in \{0,1\}$, and then the signature $\sigma = h^x$.

Verification

- Given v, a message m, the signature σ, calculates $h = H(m)$ and justifies that $\hat{e}(\sigma, g) = \hat{e}(h, v)$.

Aggregation

- Assign each user an index i for $1 < i < k$ where $k = |U|$ for the aggregating subset of users U. Each user $u_i \in U$ computes a signature $\sigma_i \in G_1$ for every message $m_i \in \{0, 1\}$ where all m_i's are their choice and all distinct. And computes the aggregate signature as $\sigma = \prod_{i=1}^{k} \sigma_i$.

Aggregate Verification

- Given the public keys v_i's, messages m_i's, the aggregate signature σ for all participants $u_i \in U$, to justify the aggregate signature,
  - makes sure of the messages m_i's are all separated from each other. Otherwise, reject; and
  - computes $h_i = H(m_i)$ for $1 < i < k$ and accepts if $\hat{e}(\sigma, g) = \prod_{i=1}^{k} \hat{e}(h_i, v_i)$.

To design the protocols for threshold, multisignature and blind signatures, one can also use the BLS Signature Scheme. [3]

3.3 Part III: Key Agreement

One of the basic cryptographic primitives is key agreement. When one needs to securely exchange some data over with someone, they first need to create a shared key. And they need to make sure its security even over an unsecure channel in case of intercepting by an adversary who want to access the message.

3.3.1 Joux's One Round 3-Party Key Agreement Scheme

To solve such problem, Diffie-Hellman protocol is an efficient way of setting a common secret key among the participants. There are some findings of setting a common secret key between more than two users; however, they all require at least two round of communication. Joux [16] showed us how to implement one round tripartite key aggreement protocol by using bilinear mappings, namely Well and Tate pairings. In this part, we will examine this protocol.

Tha aim here is to build an analog of the Diffie–Hellman protocol among three participants, A, B and C, which consists of only one round communication and end up with a common secret K_{ABC}.

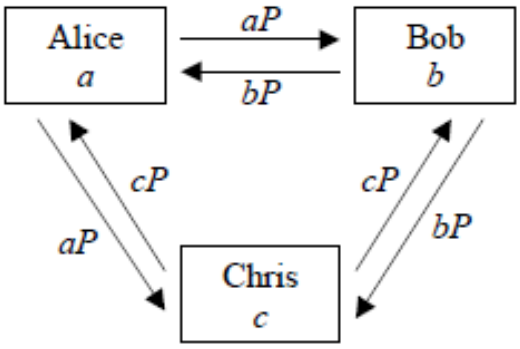

Figure 3.2: One Round 3-Party Key Agreement Protocol [16]

Let $\hat{e} : G_1 \times G_1 \rightarrow G_2$ and P as a generator of G_1. And The participants Alice, Bob and Chris have their own secrets $a, b, c \in \mathbb{Z}_q^*$, respectively. Then, the scheme is as follows:

- Alice calculates aP and broadcast it to both Bob and Chris
  Bob calculates bP and broadcast it to both Alice and Chris
  Chris calculates cP and broadcast it to both Alice and Bob.

  Please remember that these above broadcastings are done in one round of parallel message exchanges.

- By using bilinear mapping,
  Alice calculates $\hat{e}(bP, cP)^a = \hat{e}(P, P)^{abc}$
  Bob calculates $\hat{e}(aP, cP)^b = \hat{e}(P, P)^{abc}$
  Chris calculates $\hat{e}(aP, bP)^c = \hat{e}(P, P)^{abc}$

  Please remember that these calculations are computed in parallel. And now, all the participants obtain the same key $K_{abc} = \hat{e}(P, P)^{abc} \in G_2$.

  For this Diffie-Hellman based key agreement protocol, the security relies on the difficulty of the DLP on the chosen elliptic curve and in the finite field F. The elliptic curve choice is also important for the security.

3.3.2 Extending Joux's Protocol to Multi Party Key Agreement

In this protocol, Barua et al [1] uses ternary three structure, tree based group key agreement and combines them with Joux's protocol which is mentioned above. In Joux's algorithms, he uses two participants and three participants who want to create a common shared key. Barua uses this idea and instead of using only two or three participants, he uses two and three groups, namely CombineThree and CombineTwo as Diffie-Hellman key agreement protocol. There are two versions of the protocols: authenticated and unauthenticated key agreement. In the schemes below, the boxed parts are only for authenticated agreement protocol and the rest is valid for both versions.

CombineThree$(V[1,2,3], v[1,2,3])$
Consider $V[1,2,3]$ as three users set and $v[1,2,3]$ as their private keys, respectively. Also, let $Rep(V_i)$ be the representer of the user set $V[1,2,3]$ where $i \in \{1,2,3\}$.

- $Rep(V_i)$ computes $P_i = v_i P$ for each $i \in \{1,2,3\}$.
- and for the authenticated version, also computes $\boxed{T_{Rep(V_i)} = \hat{H}(P_i) v_{Rep(V_i)+v_i P_i}}$
- $Rep(V_i)$ sends P_i and $\boxed{T_{Rep(V_i)}}$ to all members of both V_i and V_k for $\{j,k\} = \{1,2,3\} \backslash i$.

  Then,

- for $\{j,k\} = \{1,2,3\} \backslash i$, each member of V_i verifies

  $$\boxed{\begin{array}{l}\hat{e}(T_{Rep(V_j)} + T_{Rep(V_k)}, P) = \\ \hat{e}(\hat{H}(P_j) Q_{Rep(V_j)} + \hat{H}(P_k) Q_{Rep(V_k)}, P_{pub}) \hat{e}(P_j, P_j) \hat{e}(P_k, P_k)\end{array}}$$

  and calculates $H(\hat{e}(P_j, P_k)^{v_i})$.

In CombineThree protocol, $H(e(P,P)^{v_1 v_2 v_3})$ is the common agreed key of three users sets.

CombineTwo$(V[1,2], v[1,2])$

Similarly to CombineThree, consider $V[1,2]$ as two users set and $v[1,2]$ as their private keys respectively. Also, let $Rep(V_i)$ be the representer of the user set $V[1,2]$ where $i \in \{1,2\}$.

- $Rep(V_i)$ computes $P_i = v_i P$ for each $i \in \{1,2\}$.
- and for the authenticated version, also computes
  $\boxed{T_{Rep(V_i)} = \hat{H}(P_i) S_{Rep(V_i)+v_i P_i}}$
- $Rep(V_1)$ generates a random $\bar{v} \in \mathbb{Z}_q^*$ and delivers $\bar{v}P$
  $\boxed{\text{and } \bar{T}_{Rep(V_1)} = \hat{H}(\bar{v}P) S_{Rep(V_1)} + \bar{v}^2 P}$ to other users.

  $\boxed{\text{every member of } V_1 \text{ and } V_2, \text{ apart from } Rep(V_1), \text{ checks: } \hat{e}(\bar{T}_{Rep(V_1)}, P) = \hat{e}(\hat{H}(\bar{v}P) Q_{Rep(V_1)}, P_{pub}) \hat{e}(\bar{v}P, \bar{v}P)}$
- $Rep(V_1)$ delivers P_1, $\boxed{T_{Rep(V_1)}}$ to every member of V_2.
  $Rep(V_2)$ delivers P_2, $\boxed{T_{Rep(V_2)}}$ to every member of V_1.
- every member of V_1 checks
  $\boxed{\hat{e}(T_{Rep(V_2)}, P) = \hat{e}(\hat{H}(P_2) Q_{Rep(V_2)}, P_{pub}) \hat{e}(P_2, P_2) \text{ and computes } H(\hat{e}(P_2, \bar{v}P)^{v_1})}$
- every member of V_2 checks
  $\boxed{\hat{e}(T_{Rep(V_1)}, P) = \hat{e}(\hat{H}(P_1) Q_{Rep(V_1)}, P_{pub}) \hat{e}(P_1, P_1) \text{ and computes } H(\hat{e}(P_1, \bar{v}P)^{v_2})}$

In CombineTwo protocol, $H(e(P,P)^{v_1 v_2 \bar{v}})$ is the common agreed key of the user sets of two.

CHAPTER 4

IMPORTANT NOTES ON PAIRINGS

While proposing a cryptographic scheme/ protocol, there are some pioneer questions that come to mind. What advantages does it have? What provides the security? Or to what kind of attacks can it be vulnarable? What should we do to keep it on the desired security level? In this chapter, you will see some notes that have been already said about these questions.

4.1 Advantages of Pairings

Basically, the pairings are classified into two types regarding protocols:

- They are used to construct the methods which can also be constructed by using other techniques. In this way, the aim is to gain efficiency.
- They are used to construct the methods which there are no other techniques to construct.

4.2 Using the Right Pairing Type

As we stated in the Preliminaries chapter, there are four types of bilinear mappings. Among these types, using the Type-1 bilinear maps (namely symmetric pairings) shows fatal security issues for the cryptographic protocols since it makes easier to attack the protocols. Nonetheless, there are still some protocols using Type-1 by

converting them into Type-3 with the help of automated tools, or in a same way, using Type-2 by converting them into Type-3. Instead of using these automated tools, to reach the desired security level and efficiency, it is recommended to use directly Type-3 while designing protocols. [18]

Although these protocols are not mentioned in this thesis, one, who wonders how type-1 and type-2 are used, can look at the schemes in the articles [14], [19] and [9]. These are some examples of wrong usage of bilinear mappings types.

4.3 Choosing the right Elliptic Curves

When implementing cryptographic protocols, the security is considered to be contingent on some cryptographic hard problems. However, while building a scheme, the intractability of these hard problems can be reduced. MOV(Menezes, Okamoto and Vanstone) [21] and FR(Frey and Ruch) reduction [12] are two examples for this case. They use bilinearity of the pairing to reduce. The basic idea of MOV reduction is as follows:

Let $\hat{e}$ be a Weil Pairing and n be the order of P, a point on an elliptic curve. And let Q be a point of order n such that there is no m where $Q = mP$, meaning P and Q are lineraly independent to each other.

Then, one can compute $\hat{e}(P, Q)$ and $\hat{e}(xP, Q) = \hat{e}(P, Q)^x$ which are now both elements of an finite field. In this way, the DLP on elliptic curves are reduced the DLP on finite fields which makes subexponential attacks possible such as MOV attacks [1].

To avoid this attack, instead of using elliptic curves with small embedding degrees, standardized safe curves should be used. The embedding degree is denoted by k and it is defined as follows:

Definition 4.1. Let E be an elliptic curve defined over $\mathbb{F}_q$, and let $P \in E(\mathbb{F}_q)$ be a point with $\gcd(n, q) = 1$ where the order of P is prime n. Then, the embedding degree of $\langle P \rangle$ is the smallest positive integer k such that $n | q^k - 1$.

[1] https://crypto.stanford.edu/pbc/notes/elliptic/movattack.html

In other words, the difficulty of DLP in $\mathbb{F}_{q^k}$ depends on choosing the embedding degree sufficiently large enough. The possibility of the known subexponential attacks for solving DLP in finite fields arises if the embedding degree k is chosen small. [20]

Also, we don't know which elliptic curves are secure. There are many weak elliptic curves. Choosing a random curve would also increase the risk for the security which is not wanted. Under these circumstances, since generating a new elliptic requires lots of works and accomplishing it rightly is difficult, it is recommended to use the standard curves.

For the Joux's key agreement scheme [16], it is recommended that p should be a 152 bits prime for $k = 2$, supersingular case.

4.4 Using Practisable Security Assumptions

Before starting to design cryptographic protocols, the security assumptions should be made very clear about the security levels. Like a butterfly effect, every choice made during the process has an impact on the security level. To reach the desired security level, for each possible pairing, elliptic curve, embedding degree, the complexity of the operations over the finite field, and so on, the security level should be found and stated. Later, the protocols should be continued with the best choice among them.

CHAPTER 5

CONCLUSION

In this thesis, we have seen some of the cryptographic protocols using pairings. It is quite amazing how far the usage of pairings have changed in the history of cryptography. Its usage area has moved up from attacking the protocols to designing new and efficient ones. To understand how pairings are used, we have seen the basic schemes of some pioneer protocols according to their usage area from encryption to signature and key agreement. Regarding how pairings are used in these systems, how much they are helpful and even in some cases how they are inalienable for the systems, we should focus on this area more than ever. That's why we believe that the interest in pairings will continue to grow even larger. We have also given some background related to pairings. As it is seen in the last chapter, to completely understand pairings and use them correctly are very crucial for security. For future work, we can

- analyze the complexitiy of the protocols using pairings and try to find new ways to lower them.
- examine the stantardized elliptic curves and try to find new elliptic curves that should be safe to use in pairing-based cryptography.

REFERENCES

[1] R. Barua, R. Dutta, and P. Sarkar, Extending joux's protocol to multi party key agreement, in *International Conference on Cryptology in India*, pp. 205–217, Springer, 2003.

[2] A. Biryukov, *Adaptive Chosen Ciphertext Attack*, pp. 21–21, Springer US, Boston, MA, 2011, ISBN 978-1-4419-5906-5.

[3] A. Boldyreva, Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme, in *International Workshop on Public Key Cryptography*, pp. 31–46, Springer, 2003.

[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, Public key encryption with keyword search, in *International conference on the theory and applications of cryptographic techniques*, pp. 506–522, Springer, 2004.

[5] D. Boneh and M. Franklin, Identity-based encryption from the weil pairing, in *Annual international cryptology conference*, pp. 213–229, Springer, 2001.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps, in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 416–432, Springer, 2003.

[7] D. Boneh, B. Lynn, and H. Shacham, Short signatures from the weil pairing, in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 514–532, Springer, 2001.

[8] R. Canetti and R. Rivest, Pairing-based cryptography, Special Topics in Cryptography Instructors: Ran Canetti and Ron Rivest Lecture, 25, 2004.

[9] J. H. Cheon, Y. Kim, H. Yoon, et al., A new id-based signature with batch verification., IACR Cryptology EPrint Archive, 2004, p. 131, 2004.

[10] R. Dutta, R. Barua, and P. Sarkar, Pairing-based cryptographic protocols: A survey., IACR Cryptology ePrint Archive, 2004, p. 64, 2004.

[11] N. El Mrabet and M. Joye, *Guide to pairing-based cryptography*, Chapman and Hall/CRC, 2017.

[12] G. Frey and H.-G. Rück, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, Mathematics of computation, 62(206), pp. 865–874, 1994.

[13] S. D. Galbraith, K. G. Paterson, and N. P. Smart, Pairings for cryptographers, Discrete Applied Mathematics, 156(16), pp. 3113–3121, 2008.

[14] Y. Gong, Y. Cai, Y. Guo, and Y. Fang, A privacy-preserving scheme for incentive-based demand response in the smart grid, IEEE Transactions on Smart Grid, 7(3), pp. 1304–1313, 2015.

[15] D. Hankerson, A. J. Menezes, and S. Vanstone, Guide to elliptic curve cryptography, Computing Reviews, 46(1), p. 13, 2005.

[16] A. Joux, A one round protocol for tripartite diffie–hellman, in *International algorithmic number theory symposium*, pp. 385–393, Springer, 2000.

[17] J. Katz, A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*, CRC press, 1996.

[18] M. S. Kiraz and O. Uzunkol, Still wrong use of pairings in cryptography, arXiv preprint arXiv:1603.02826, 2016.

[19] F. Li, Z. Zheng, and C. Jin, Identity-based deniable authenticated encryption and its application to e-mail system, Telecommunication Systems, 62(4), pp. 625–639, 2016.

[20] A. Menezes, An introduction to pairing-based cryptography, Recent trends in cryptography, 477, pp. 47–65, 2009.

[21] A. J. Menezes, T. Okamoto, and S. A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, iEEE Transactions on information Theory, 39(5), pp. 1639–1646, 1993.

[22] J. H. Silverman, *The arithmetic of elliptic curves*, volume 106, Springer Science & Business Media, 2009.

[23] J. Tate, Wc-groups over p-adic fields, Séminaire Bourbaki, 156, pp. 156–1, 1957.