A PAIRING-FREE IDENTITY-BASED CRYPTOSYSTEM

Poonsuk Ponpurmpoon

A Dissertation Submitted in Partial
Fulfillment of the Requirements for the Degree of
Doctor of Philosophy (Computer Science and Information Systems)
School of Applied Statistics
National Institute of Development Administration
2021

A PAIRING-FREE IDENTITY-BASED CRYPTOSYSTEM
Poonsuk Ponpurmpoon
School of Applied Statistics

..... Major Advisor
(Associate Professor Ohm Sornil, Ph.D.)

..... Co-Advisor
(Associate Professor Pipat Hiranvanichakorn, D.Eng.)

The Examining Committee Approved This Dissertation Submitted in Partial Fulfillment of Requirements for the Degree of Doctor of Philosophy (Computer Science and Information Systems).

..... Committee Chairperson
(Associate Professor Surapong Auwatanamongkol, Ph.D.)

..... Committee
(Associate Professor Pipat Hiranvanichakorn, D.Eng.)

..... Committee
(Associate Professor Ohm Sornil, Ph.D.)

..... Committee
(Norranut Saguansakdiyotin, Ph.D.)

..... Dean
(Assistant Professor Pramote Luenam, Ph.D.)

_____/_____/_____

ABSTRACT

Title of Dissertation A PAIRING-FREE IDENTITY-BASED CRYPTOSYSTEM
Author Poonsuk Ponpurmpoon
Degree Doctor of Philosophy (Computer Science and Information Systems)
Year 2021

ID-based cryptosystems (IBCs) allow the use of publicly identifiable information in public encryption keys, which reduces the overhead of certificate management and eliminates the need for a certificate authority in the public-key infrastructure. Up to now, bilinear pairing technology is usually used in ID-based paradigms. However, it is expensive in computation time and is unsuitable for mobile networks. Over recent years, the evolution of mobile devices has seen them transformed from a voice communication device to a daily life information center that is restricted by poor battery capacity and limited computation power. Thus, interest in pairing-free ID-based algorithms among researchers is growing. Herein, a pairing-free IBC consisting of ID-based encryption, digital signatures, and key exchange schemes is presented. All of the schemes use the same public and private key definitions, which makes IBC implementation straightforward. Proof of the correctness and security analysis of the scheme are provided. Furthermore, the performance of the proposed system is compared with well-known pairing-based systems and other well-known pairing-free ones.

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to Associate Professor Pipat Hiranvanichakorn for his insightful comments and continuous support. This thesis could not have been completed without his guidance.

My sincere thanks also go to my thesis committee: Associate Professor Surapong Auwatanamongkol, Associate Professor Ohm Sornil, and Ajarn Norranut Saguansakdiyotin for their insightful comments and helpful suggestions, encouragement, and follow-up on the progress of this dissertation.

I wish to thank the NIDA staff for their hard work, dedication, and devotion toward student achievement. Thank you for your excellent support and helpfulness.

Last but not least, I am extremely grateful to my parents for their constant support and endless patience while completing this dissertation.

Poonsuk Ponpurmpoon

December 2021

TABLE OF CONTENTS

Page

LIST OF TABLES

Tables **Page**

LIST OF FIGURES

Figures **Page**

CHAPTER 1

INTRODUCTION

In 1984, (Shamir, 1985) introduced the concept of the asymmetric-key ID-based cryptosystem (IBC) paradigm. In theory, the advantages of IBCs are that they allow the use of publicly identifiable information as public keys, which reduces the overhead cost of certificate management and gets rid of the need for a certificate authority (CA) in the public-key infrastructure. Public keys based on the user's identity are a meaningful and true reflection of the user's personality, while the user's public keys in non-IBCs are mathematically computed numbers.

Nowadays, the IBC concept is applied in many areas (Hess, 2003; Hölbl, Welzer, & Brumen, 2012; Islam & Biswas, 2011; A. Kumar, Tripathi, & Jaiswal, 2015; M. Kumar, Katti, & Saxena, 2017; Li, Dai, & Yang, 2011; Malina, Hajny, & Zeman, 2015; Ming & Yuan, 2019; Nathani, Tripathi, & Khatoon, 2019; Zhandry, 2012; Zhu, Yang, & Wong, 2005). In paper (Li et al., 2011), ID-based encryption (IBE) and ID-based signature (IBS) have been used for encrypting and signing messages between users in a hierarchical architecture for cloud computing (HACC) scenario. The identity of the user is defined by nodes in the hierarchical structure from the user's node to the root node of the HACC. IBC has been utilized to generate the user's public key from his/her identity in an ID-based blind signature approach for E-voting(M. Kumar et al., 2017); IBC was utilized because it has the merit that the voter's public key is directly derived from his/her identity. IBC has been used to implement a dynamic authenticated group key agreement in (Nathani et al., 2019). In (Ming & Yuan, 2019), anonymous ID-based broadcast encryption with asymmetric bilinear pairing was implemented.

In 2000, (Joux, 2000) proposed the one-round three-party key agreement protocol that offers the potential of pairing with an ID-based paradigm. In 2001, (Boneh & Franklin, 2003) introduced the first implementation of an IBE scheme by applying bilinear pairing on an elliptic curve for an IBC. Later, (Smart, 2002)

introduced the ID-based two-party key agreement protocol based on Weil pairing. However, as mentioned by (Jin, Debiao, & Jianhua, 2010), bilinear pairing is expensive in computation time and is considered to be unsuitable for mobile network computing. As (Liu, Cao, Kong, & Wang, 2017) pointed out, bilinear groups with a large composite order do not provide substantial benefit to cryptographic schemes in practice. There is another important issue for an ID-based pairing algorithm. Since the public key from an IBC is derived from the user's identity, if the user's private key is compromised, it is not easy for the key generation center (KGC) to change the user's private and public key pairs. Thus, interest in pairing-free ID-based algorithms is growing, and several approaches to implement them have been proposed.

Many approaches using elliptic curve cryptography (ECC) (Cao, Kou, & Du, 2010; Islam & Biswas, 2017; Jin et al., 2010; Koblitz, Koblitz, & Menezes, 2011; Abhimanyu Kumar & Tripathi, 2016; Naresh & Murthy, 2015; Roy & Khatwani, 2017; Sakai & Kasahara, 2003) have been suggested for ID-based pairing-free cryptosystems. In a digital signature scheme, (Jin et al., 2010) suggested a protocol based on the elliptic curve digital signature algorithm (ECDSA) (American National Standards Institute, 1998). They claimed that the computational time for the proposed protocol was lower than bilinear pairing ones. The disadvantage of this protocol is that the user's private key is not confidential because a part of the private key has to be sent out for use in the signature verification process. Nevertheless, some pairing-free key agreement schemes have been proposed (Cao et al., 2010; Chakraborty, Raghuraman, & Rangan, 2016; Islam & Biswas, 2017; Abhimanyu Kumar & Tripathi, 2015, 2016; Xiaozhuo, Taizhong, Weihua, & Yongming, 2014). In the approach of (Abhimanyu Kumar & Tripathi, 2016), the key exchange parties must send out some parts of the private key to establish a shared key, and thus the private key is not confidential, and the scope of the research was also limited to parties that are members of the same KGC. However, in practice, shared keys established for users belonging to different KGCs are needed.

A pairing-free IBE algorithm has not yet been established, which may be due to the confidentiality problem in the use of the user's private key, as discussed above. Moreover, in previous ID-based pairing-free systems comprising digital signatures

and key agreement, researchers have often defined the system parameters and private key extraction mechanism to fit their proposed protocols. Indeed, the use of the general system parameter and key definitions applied in the ID-based digital signature, encryption, and key exchange schemes in a cryptosystem to make its implementation simple has not previously been reported.

In this thesis, a pairing-free IBC consisting of IBE, ID-based digital signature, and ID-based key exchange schemes is proposed. All of the schemes use the same public and private key definitions that have two advantages to address security concerns. First, the definitions can prevent other parties from discovering the KGC master key. Second, if the user's private key is compromised, the KGC can easily generate a new one. As for ID-based key exchange, the proposed system can cope with system parameters from different KGCs, which means that the users can be on different KGCs. This non-shared system parameter system is useful for mobile network computing in real scenarios. We conducted a security analysis of the proposed system to address security concerns. The results show that it is durable to several types of attacks, such as man-in-the-middle, which can occur during the encryption-decryption and key exchange processes, and fake signatures, which can occur in the digital signature creation-verification process. Furthermore, we compared our proposed pairing-free scheme with some well-known pairing-free and pairing-based ones.

The remainder of this thesis is organized as follows. In chapter 2, we introduce the basic concepts of IBC, ECC, and bilinear pairing in cryptography. At the end of chapter 2, some examples of pairing IBE algorithms, signature algorithms, and key exchange are offered. In chapter 3, some examples of pairing-free algorithms and their security analyses are given. In chapter 4, the proposed pairing-free algorithm and proof of its correctness are presented. The security and performance analysis of the proposed cryptosystem is discussed in chapter 5. Finally, conclusions on the study are offered in chapter 6.

CHAPTER 2

BACKGROUND FOR THE STUDY

In this chapter, four topics that cover the background for the study are briefly introduced. First, the encryption and signature methods in ID-based cryptosystems (IBCs) are described followed by an overview of elliptic curve cryptography (ECC). Next, the concept of bilinear pairing along with some of its properties is covered. Finally, some pairing-based algorithms are reviewed.

2.1 ID-Based Cryptosystems (IBCs)

IBCs first came to light in 1984 when (Shamir, 1985) proposed a system that uses the user's identity information (e.g., email address, name, phone number, etc.) as a public key. This information is publicly known, so it does not require a certificate authority (CA) to certify the key, which means that a special process to broadcast, store, and maintain the key is not required. In IBCs, a trusted authority such as a key generation center (KGC) generates the user's private key by using the user's public key and the KGC's private key. The KGC then sends the private key to the corresponding user via a secure channel.

2.1.1 ID-Based Encryption (IBE)

In asymmetric-key cryptography, a message is encrypted by using the authenticated public key of the receiver, and the mathematical-related private key of the receiver is used to decrypt the message. In IBE, a message is encrypted by using the receiver's ID and the KGC public key. A detailed description of the IBE scheme can be found in (Youngblood, 2005). A simple version of IBE and decryption is demonstrated as follows.

Ciphertext C of message M is computed as

$$C = \text{Encrypt}(\text{recipient}_{pub}, \text{KGC}_{pub}, M),$$

where recipient_{pub} is the public key of the recipient and KGC_{pub} is the KGC public key. The recipient uses his/her private key to decrypt the message. In general, the recovery process can be derived as

$$M = \text{Decrypt}(C, \text{recipient}_{pri}),$$

where recipient_{pri} is the corresponding private key of the recipient.

Figure 2.1 illustrates the idea of IBE. Alice encrypts a message with the KGC public key and Bob's public key, then she sends it to Bob. Bob authenticates himself via the key server and then retrieves his private key to decrypt Alice's message.

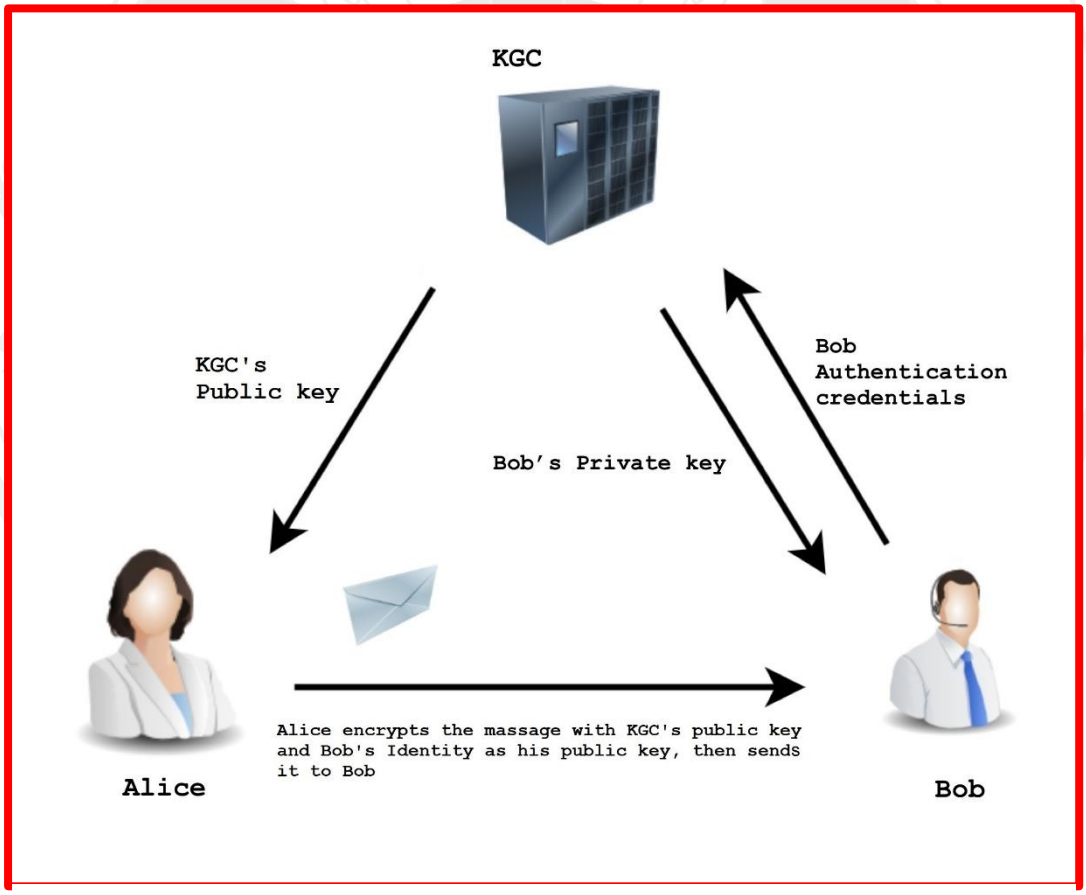

Figure 2.1 The basic concept of IBE.

2.1.2 ID-Based Signature (IBS)

A digital signature is a mathematical process to guarantee that the sender of the message cannot deny that he/she created the message, while the receiver cannot deny having received the message. It is created by applying the message to the signing algorithm.

Figure 2.2 illustrates the basic concept of IBS. Alice authenticates herself with the KGC and retrieves her private key. Alice then signs the message and sends the message with a signature to Bob. Bob verifies the received signature with Alice's identity via her public key and the KGC public key.

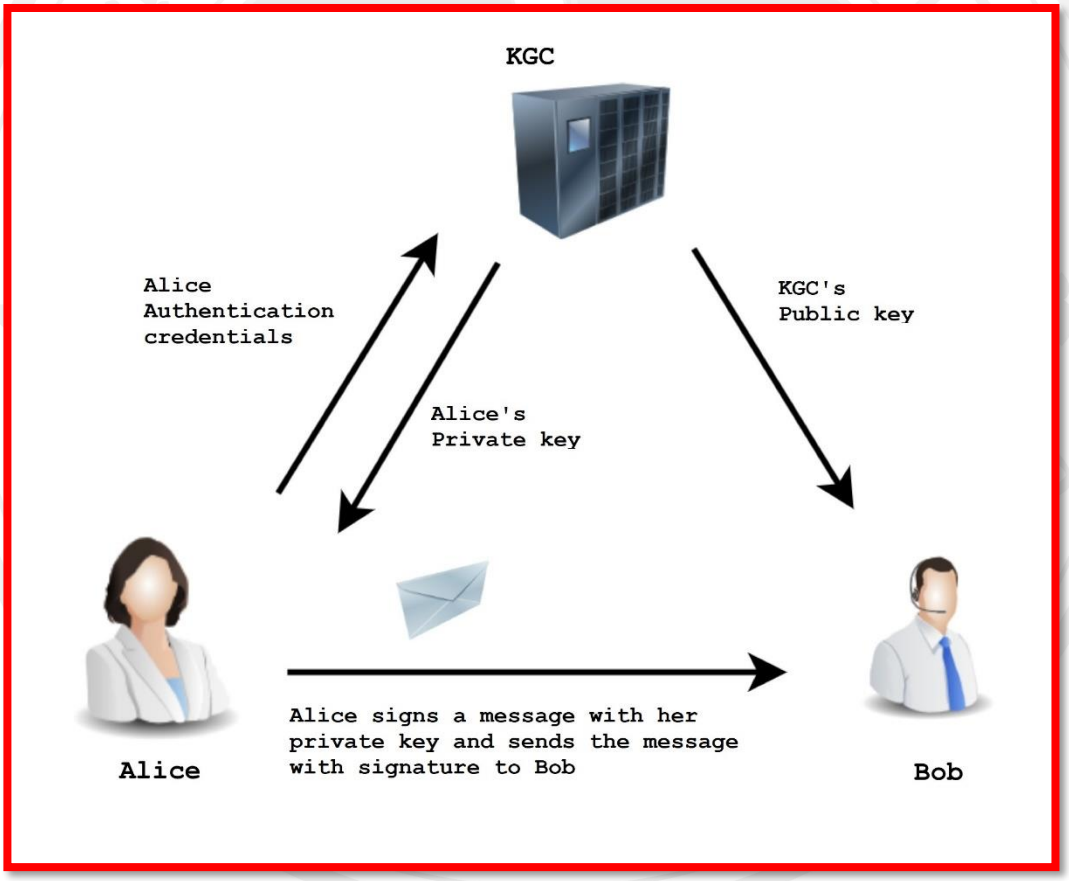

Figure 2.2 The basic concept of IBS.

Some examples of the implementation of IBS can be found in (Gentry & Silverberg, 2002). The signing and verification process is as follows:

The signer uses his/her private key to sign the message. The user's signature is computed as

S_i = Sign(*system parameters*, $signer_{pri}$, M).

As for the verification process, the verifier uses the signer's public key and the KGC public key to verify the signature. The logic for this is as follows:

If Verify(*system parameters*, KGC_{pub}, M, $signer_{pub,}$ S_i), then accepts the signature.

The advantage and disadvantages of using traditional IBC are described below.

The Advantage of ID-based Cryptography

a) Since every user's identity is known, there is no need to manage the CA.

The Disadvantages of ID-Based Cryptography

a) Since the KGC knows the private keys of all users, the decryption and signing process is performed under the KGC, thus the cryptosystem can get into problems under dishonest KGC.
b) In a pairing-based system, the user's public key is based on his/her identity, and thus it is difficult to revoke it when the user's private key has been compromised.

2.2 Elliptic Curve Cryptography (ECC)

ECC is very useful in several areas of mathematics. It is a new direction away from existing cryptosystems and plays an important role in ID-based cryptography in both pairing-enforced and pairing-free schemes (Hoffstein, Pipher, & Silverman, 2008). In this chapter, some of the concepts of ECC that are of interest in this study are introduced. In ECC, the elliptic curve is defined over finite field $\mathbb{F}_p$ in the general form

$$E: Y^2 = x^3 + ax + b,$$

where a and b are real numbers and $4a^3 + 27b^2 \neq 0$. Points on elliptic curve E and the point at infinity O, which is also an identity element in a finite Abelian group with the + operation, can be combined to produce a finite Abelian group with suitable properties. The following are some of the main properties of ECC.

2.2.1 Scalar Point Multiplication

$$kP = (P + P + P + \cdots + P)_{k \text{ times}},$$

where k is a scalar and P is a point on elliptic curve E defined by the adding it k times. Some of the complex problems to do with ECC are as follows.

2.2.2 The Elliptic Curve Discrete Logarithm Problem (ECDLP)

Let E be an elliptic curve over finite field $\mathbb{F}_p$ and P be a point on elliptic curve E. For a given kP, find integer k. In ECDLP, kP is relatively easy to compute. However, computing k is intractable even when P and kP are known. In this problem, k is usually used as the user's private key while kP is used as the corresponding public key.

2.2.3 The Elliptic Curve Diffie-Hellman Problem (ECDHP)

Let E be an elliptic curve over finite field $\mathbb{F}_p$, P be a point on elliptic curve E, and a and b be integers. For a given $A = aP$, $B = bP$, find point $C = abP$.

In ECDHP, although it is relatively easy to compute C when P, a, and B (or P, b, and A) are available, exponential time is needed to compute abP even when P, A, and B are known. In this problem, abP is usually used as the shared key between users A and B.

2.3 Bilinear Pairing in Cryptography

In this section, the basic theory of bilinear pairing in cryptography is described, after which the properties of bilinear pairing are addressed. An overview of research into bilinear pairing can be found in (Hoffstein et al., 2008).

Let $\mathbb{G}_1$ be a cyclic additive group and $\mathbb{G}_2$ be a cyclic multiplicative group. Both $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. Bilinear pairing is achieved by mapping $\hat{e}$: $\mathbb{G}_1$ x $\mathbb{G}_1$ -> $\mathbb{G}_2$, which satisfies the following properties.

Computability. For all U, $V \in \mathbb{G}_1$, there is an algorithm to efficiently compute $\hat{e}$: (U, V).

Non-degeneracy**.** There exists $P \in \mathbb{G}_1$, such that $\hat{e}$: $(P, P) \neq 1$.

Bilinearity. For all $U, V, W \in \mathbb{G}_1$,

$$\hat{e}(U, V+W) = \hat{e}(U, V) \bullet \hat{e}(U, W) \text{ and}$$

$$\hat{e}(U+W, V) = \hat{e}(U, V) \bullet \hat{e}(W, V).$$

For $a, b \in \mathbb{Z}_q$, $\hat{e}(aU, bV) = \hat{e}(U, bV)^a = \hat{e}(aU, V)^b$

$$= \hat{e}(U, V)^{ab} = \hat{e}(abU, V) = \hat{e}(U, abV).$$

2.4 Examples of Pairing-Based IBCs

In this section, some examples of pairing-based IBCs to compare the performances of pairing-based algorithms and the proposed pairing-free ones are provided. The notation used in the pairing-based schemes is defined in Table 2.1.

Table 2.1 The Notation Used in Pairing-Based Schemes.

Notation	Type	Description
$\mathbb{G}_1$	An additive group of points on E/ $\mathbb{F}_p$	
$\mathbb{G}_2$	A multiplicative group of finite fields $\mathbb{F}_{p^2}^*$	
sk_1	Integer	The KGC master key
P	A point on the elliptic curve	The group generator of $\mathbb{G}_1$
Q_{ID}	A point on the elliptic curve	The hash value of the user's ID
S_{ID}	A point on the elliptic curve	The user's private key
P_{pub}	A point on the elliptic curve	The KGC public key
(R,Z)	A pair of points on the elliptic curve	Signature (R,Z) of the user
K_A^B, K_B^A	$\in \mathbb{G}_2$	The shared key between users A and B

2.4.1 IBE Algorithms

The first practical IBE scheme on pairing was proposed by (Boneh & Franklin, 2003). Their encryption scheme applied pairing to compute the symmetric

component between the sender and the recipient. In this algorithm, $sk_1 \in \mathbb{Z}_q^*$ is the master key from the KGC, random number $r \in \mathbb{Z}_q^*$, P is the group generator of $\mathbb{G}_1$, $\mathbb{G}_1$ is an additive group of points on E/$\mathbb{F}_p$, $\mathbb{G}_2$ is a multiplicative group of finite field $\mathbb{F}_{p^2}^*$. H_1: $(0,1)^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \rightarrow (0,1)^n$. $Q_{ID} = H_1(\text{ID})$. The user's private key $S_{ID} = sk_1 \bullet Q_{ID}$ and KGC's public key $P_{pub} = sk_1 \bullet P$, $g_{ID} = \hat{e}(P_{pub}, Q_{ID}) \in \mathbb{G}_2^*$.

Ciphertext C = $(r \bullet P, \text{M} \oplus H_2(g_{ID}{}^r))$.

Decrypted message $M = M \oplus H_2(g_{ID}{}^r) \oplus H_2(\hat{e}(r \bullet P, S_{ID}))$

$= M \oplus H_2(g_{ID}{}^r) \oplus H_2(\hat{e}(r \bullet P, sk_1 \bullet Q_{ID}))$

$= M \oplus H_2(g_{ID}{}^r) \oplus H_2(\hat{e}(P, Q_{ID})^{r \bullet sk_1})$ (from $\hat{e}(aU, bV) = \hat{e}(U, V)^{ab}$)

$= M \oplus H_2(g_{ID}{}^r) \oplus H_2(\hat{e}(sk_1 \bullet P, Q_{ID})^r)$ (from $\hat{e}(U, V)^{ab} = \hat{e}(aU, V)^b$)

$= M \oplus H_2(g_{ID}{}^r) \oplus H_2(\hat{e}(P_{pub}, Q_{ID})^r)$ (from $P_{pub} = sk_1 \bullet P$)

$= M \oplus H_2(g_{ID}{}^r) \oplus H_2(g_{ID}{}^r)$.

2.4.2 IBS Algorithms

(Paterson, 2002) proposed an ID-based digital signature scheme based on bilinear pairing. The pairing is utilized to validate the authenticity of signature *(R, Z)* for message *M*. In this algorithm $k \in \mathbb{Z}_q^*$ is a random integer and P is the group generator of $\mathbb{G}_1$. $H_3 : (0,1)^* \rightarrow \mathbb{Z}_q$, $H_4 : \mathbb{G}_1 \rightarrow \mathbb{Z}_q$, $Q_{ID} = H_1(\text{ID})$, $S_{ID} = sk_1 \bullet Q_{ID}$, $sk_1 \in \mathbb{Z}_q$ is the master key of KGC, and $P_{pub} = sk_1 \bullet P$ is the corresponding public key of KGC.

Signature (*R*,*Z*) is calculated as

$R = k \bullet P$,

$Z = k^{-1}(H_3(m) \bullet P + H_4(R) \bullet S_{ID})$.

The verifier can verify received signature *(R,Z)* by computing $\hat{e}(\text{R},Z)$ and $\hat{e}(P,P)^{H_3(\text{m})} \bullet \hat{e}(sk_1 P, Q_{ID})^{H_4(\text{R})}$. If $\hat{e}(\text{R},Z) = \hat{e}(P,P)^{H_3(\text{m})} \bullet \hat{e}(sk_1 P, Q_{ID})^{H_4(\text{R})}$, then received signature (*R*,*Z*) is genuine. The following computation shows how bilinearity can be used to verify received signature (*R*,*Z*):

$\hat{e}(R,Z) = \hat{e}(k \bullet P, k^{-1}(H_3(m) \bullet P + H_4(R) \bullet S_{ID}))$

$= \hat{e}(P, (H_3(m) \bullet P + H_4(R) \bullet S_{ID}))^{k \bullet k^{-1}}$ (from $\hat{e}(aU, bV) = \hat{e}(U, V)^{ab}$

$= \hat{e}(P, (H_3(m) \bullet P + H_4(R) \bullet S_{ID}))$

$= \hat{e}(P, (H_3(m) \bullet P)) \bullet \hat{e}(P, H_4(R) \bullet S_{ID})$ (from $\hat{e}(U,V+W) = \hat{e}(U,V) \bullet \hat{e}(U,W)$)

$= \hat{e}(P, H_3(m) \bullet P) \bullet \hat{e}(P, \bullet S_{ID})^{H_4(R)}$ (from $\hat{e}(aU, bV) = \hat{e}(aU, V)^b$)

$= \hat{e}(P, H_3(m) \bullet P) \bullet \hat{e}(P, \mathrm{sk}_1 \bullet Q_{ID})^{H_4(R)}$

$= \hat{e}(P, P)^{H_3(m)} \bullet \hat{e}(P, Q_{ID})^{H_4(R) \bullet sk_1}$ (from $\hat{e}(aU, bV) = \hat{e}(aU, V)^b$)

$= \hat{e}(P, P)^{H_3(m)} \bullet \hat{e}(sk_1 \bullet P, Q_{ID})^{H_4(R)}$ (from $\hat{e}(U, V)^{ab} = \hat{e}(aU, V)^b$).

2.4.3 ID-Based Key Exchange

This is the last part of an IBC that uses a pairing technique. (Smart, 2002) implemented the first ID-based key exchange protocol based on pairing. To obtain a shared key, $K_A^B = \hat{e}(aQ_B, P_{KGC}) \bullet \hat{e}(S_A, T_B)$ is computed for user A and $K_B^A = \hat{e}(S_B, T_A) \bullet \hat{e}(bQ_A, P_{KGC})$ is computed for user B.

In this algorithm, sk_1 is the secret key from the KGC; a and b are the session private keys of users A and B, respectively; $Q_{ID} = H_1(ID)$; the long-term private keys of the users are $S_A = sk_1 Q_A$ and $S_B = sk_1 Q_B$; and the session public keys of the users are $T_A = aP$ and $T_B = bP$.

The following proof shows that $K_A^B = K_B^A$.

$K_A^B = \hat{e}(aQ_B, P_{KGC}) \bullet \hat{e}(S_A, T_B)$

$= \hat{e}(aQ_B, sk_1 \bullet P) \bullet \hat{e}(sk_1 \bullet Q_A, bP)$

$= \hat{e}(Q_B, P)^{sk_1 \bullet a} \bullet \hat{e}(Q_A, P)^{sk_1 \bullet b}$ (from $\hat{e}(aU, bV) = \hat{e}(U, V)^{ab}$),

$= \hat{e}(sk_1 Q_B, aP) \bullet \hat{e}(bQ_A, sk_1 \bullet P)$ (from $\hat{e}(U, V)^{ab} = \hat{e}(aU, bV)$),

$= \hat{e}(S_B, T_A) \bullet \hat{e}(bQ_A, P_{KGC})$

$= \hat{e}(bQ_A, P_{KGC}) \bullet \hat{e}(S_B, T_A)$

$= K_B^A$.

Another ID-based key exchange example can be found in (Lee & Lee, 2005). In this algorithm, sk_1 is the secret key of the KGC; $Q_{ID} = H_3(ID)$; the private keys of users A and B are $S_A = sk_1 Q_A$ and $S_B = sk_1 Q_B$, respectively; and $kdf : \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \{0,1\}^*$, where kdf is the key derivation function. User A randomizes a as a short-term private key, computes short-term public key $V_A = aQ_B$ and $W_A = aS_A$, and then sends

(V_A,W_A) to user *B*. Meanwhile, user *B* randomizes *b* as a short-term private key, computes short-term public key $V_B = bQ_A$ and $W_B = bS_B$, and then sends (V_B,W_B) to user *A*. These actions are summarized as follows:

$A \rightarrow B$: (V_A,W_A),

$B \rightarrow A$: (V_B,W_B).

User *A* then computes $K_A^B = \hat{e}(aQ_A + V_B, W_B)^a$ while user *B* computes $K_B^A = \hat{e}(bQ_B + V_A, W_A)^b$. Finally, users A and B compute a shared common key as $K = kdf(K_A^B, Q_A, Q_B) = kdf(K_B^A, Q_A, Q_B) = kdf(\hat{e}(Q_A, Q_B)^{(a+b)absk_1}, Q_A, Q_B)$.

The following proof shows that $K_A^B = K_B^A$.

$$K_A^B = \hat{e}(aQ_A + V_B, W_B)^a$$
$$= \hat{e}(aQ_A + bQ_A, bS_B)^a$$
$$= \hat{e}((a + b)Q_A, bS_B)^a$$
$$= \hat{e}((a + b)Q_A, b\,sk_1\,Q_B)^a$$
$$= \hat{e}(Q_A, Q_B)^{(a+b)absk_1},$$

$$K_B^A = \hat{e}(bQ_B + V_A, W_A)^b$$
$$= \hat{e}(bQ_B + aQ_B, aS_A)^b$$
$$= \hat{e}((b + a)Q_B, aS_A)^b$$
$$= \hat{e}((b + a)Q_B, a\,sk_1\,Q_A)^b$$
$$= \hat{e}(Q_B, Q_A)^{(a+b)absk_1}.$$

Since $K_A^B = K_B^A = \hat{e}(Q_A, Q_B)^{(a+b)absk_1}$, then $kdf(K_A^B, Q_A, Q_B) = kdf(K_B^A, Q_A, Q_B) = kdf(\hat{e}(Q_A, Q_B)^{(a+b)absk_1}, Q_A, Q_B)$.

Although several ID-based algorithms have been based on bilinear pairing, the high computational complexity to compute these causes the computation time to be too long for actual implementation (Benhamouda, Couteau, Pointcheval, & Wee, 2015; Jin et al., 2010). Another drawback of pairing-based ID-based schemes arises when the user's private key has been compromised. In ID-based schemes, the user's public key is made up of the user's identity comprising his/her name, email address,

phone number, etc. The user's identity does not often change, which means that his/her public key seldom does either. As the user's private key is generated by the corresponding public key, the private key is not easily renewed.

For the practical implementation of IBCs, a pairing-free basis has been considered to achieve better computation time. In general, pairing-free-based algorithms can be computed three times faster than pairing-based ones (Benhamouda et al., 2015). Hence, pairing-free schemes are more suitable for devices with limited power, such as mobile devices. In the next chapter, some pairing-free ID-based algorithms and an analysis of their security are covered.

2.5 Elliptic Curve Digital Signature Algorithm (ECDSA) (Stalling, 2013)

In this algorithm, random integer $s_A \in [1, n - 1]$ is chosen as user A's private key. User A then computes $Q = s_A \cdot P_1$ as his/her public key, where P_1 is a base point on the elliptic curve of order n. Let HF_5 be the SHA-1 hash function giving a 160-bit integer value as the output. When user A wants to send a message together with a signature to user B, signature pair (r,z) is generated for message M by computing the following steps:

1. Choose a random integer $k \in [1, n - 1]$.
2. Compute point $(x,y) = k \cdot P_1$ and the first portion of the signature $r = x \bmod n$, for $r \neq 0$.
3. Compute $e = HF_5(M)$.
4. Compute $z = k^{-1}(e+s_A r) \bmod n$, for $z \neq 0$.
5. For pair (r,z) as the signature of message M:

When user B receives (r,z) from user A, the following steps are executed to verify the message:

1.Compute $e = HF_5(M)$.
2. Compute $w = z^{-1} \bmod n$.
3. Compute $u_1 = ew$ and $u_2 = rw$.
4. Compute point $(x_1,y_1) = u_1 \cdot P_1 + u_2 \cdot Q$.
5. Compute $v = x_1 \bmod n$.

6. Accept the signature (r,z) from user A if and only if $v = r$.

The following proof shows that this algorithm is accurate.

Define $z = k^{-1}(e+s_A r) \bmod n$,

then it follows that

$$k = z^{-1}(e+s_A r) \bmod n,$$
$$k = (z^{-1}e + z^{-1}s_A r) \bmod n,$$
$$k = (ew + ws_A r) \bmod n,$$
$$k = (u_1 + u_2 s_A) \bmod n.$$

For step 4 in the algorithm, (x_1,y_1) is computed as

$$\begin{aligned}(x_1,y_1) &= u_1 \cdot P_1 + u_2 \cdot Q \\ &= u_1 \cdot P_1 + u_2 \cdot (s_A \cdot P_1) \\ &= u_1 \cdot P_1 + u_2 \cdot s_A \cdot P_1 \\ &= (u_1 + u_2 \cdot s_A)\, P_1 \\ &= k \cdot P_1. \\ &= (x,y)\end{aligned}$$

Since v is computed from $(u_1 \cdot P_1 + u_2 \cdot Q)$ and r from $(k \cdot P_1)$, then $v = r$ and $(u_1 \cdot P_1 + u_2 \cdot Q) = (k \cdot P_1)$.

CHAPTER 3

SECURITY ANALYSIS OF THE CURRENT PAIRING-FREE ID-BASED SCHEMES

Two pairing-free IBCs are discussed in this chapter. The first is the ID-based digital signature without pairing scheme proposed by (Jin et al., 2010). The second one is the ID-based group key agreement without pairing scheme proposed by (Abhimanyu Kumar & Tripathi, 2016). Notably, pairing-free IBE algorithms have not previously been reported. The notation used in the pairing-free schemes described in Chapters 3 and 4 is defined in Table 3.1.

Table 3.1 Notation for the Pairing-Free Schemes.

Notation	Type	Description
sk_i	Integer	The private key of KGC_i, for i = 1, 2.
r_j	Integer	A random number generated by the KGC to compute a private key of user j, for $j = A, B, C$.
n	Integer	The order of points on the curve.
(D_j, s_j)	(A point on the curve, integer)	User j's private key used in (Jin et al., 2010) and (Abhimanyu Kumar & Tripathi, 2016), for $j = A, B, C$.
P_i	A point on the curve	A base point on the elliptic curve.
Q	A point on the curve	The user's public key defined by using the ECDSA algorithm.
$P_{pub\ i}$	A point on the curve	The public key of KGC_i
K_A^B, K_B^A	A point on the curve	The shared key between users A and B.

Notation	Type	Description
s_j	Integer	The private key of user *j*, for *j = A, B, C, X*.
$(ID_j.R_j)$	(A pair of strings, a string representing a point on the curve)	User *j's* public key, for *j = A, B, C, X*.
(r,z)	A pair of integers	Digital signature (r,z) for message *M*.
x_j	Integer	Session private key of user *j*, where *j = A, B, C, X*.

3.1 ID-Based Digital Signatures Without Pairing (Jin et al., 2010)

In this algorithm, the KGC's private key is defined as $sk_1 \in \mathbb{Z}_n^*$, while $r_A \in \mathbb{Z}_n^*$ is randomly produced to generate a private key for user *A*. The public key for user A is the user's ID and the corresponding private key is (D_A, s_A); D_A and s_A are respectively computed by the KGC as $D_A = r_A \cdot P$ and $s_A = (r_A + h \bullet sk_1) \bmod n$, where $h = HF_1 (ID_A \| D_A)$ is the hashed user's ID using hash function HF_1: $\{0,1\}^* \rightarrow \mathbb{Z}_n^*$. Note that *P* is a base point on the elliptic curve of order *n*.

At the end of the parameter initialization phase in (Jin et al., 2010), the KGC sends user A's private key (D_A, s_A) to user A through a secure channel. User A can validate the received private key (D_A, s_A) by computing $s_A \cdot P = D_A + h \cdot P_{pub}$, as shown by the following proof:

$$s_A \cdot P = D_A + h \cdot P_{pub}.$$
$$(r_A + h \cdot sk_1) \cdot P = (r_A \cdot P) + h \cdot P_{pub}$$
$$(r_A + h \cdot sk_1) \cdot P = (r_A \cdot P) + h \cdot (sk_1 \cdot P)$$
$$(r_A + h \cdot sk_1) \cdot P = (r_A + h \cdot sk_1) \cdot P.$$

The signature signing and verification process in the present study follows the ECDSA algorithm (Jin et al., 2010). In the signing process, s_A is used to sign the message from user A, and then (D_A, r(*ECDSA*), s(*ECDSA*)) is sent out as his/her message signature. r(*ECDSA*) is a random number defined in the ECDSA and s(*ECDSA*) is the digital signature computed in the ECDSA. To verify the signature, D_A, h, and P_{pub} are used by the receiver to compute $Q = D_A + h \cdot P_{pub}$, and then Q is used as user A's public key in the ECDSA algorithm.

To generate signature $(r(ECDSA), s(ECDSA))$ for message m, user A randomly selects an integer, k, and computes $kP = (x_1, y_1)$:

$$r(ECDSA) = x_1 \bmod n,$$

$$e = sha\text{-}1(m)\ ,\ \text{where sha-1 is a hash function, and}$$

$$s(ECDSA) = k^{-1} (e + s_A \cdot r(ECDSA)) \bmod n.$$

Subsequently, user A sends out $(D_A, r(ECDSA), s(ECDSA))$ as the signature of message m.

On receiving the signature $(D_A, r(ECDSA), s(ECDSA))$, the verifier computes

$$Q = D_A + h \cdot P_{pub}$$

and uses it as user A's public key to verify the signature according to the ECDSA algorithm.

The advantages of the approach in (Jin et al., 2010) are that implemented system relies on ECDSA; it has been unarguably proved that ECDSA offers strong security with efficient performance. The authors mentioned that bilinear pairing consumes a lot of computation time. Indeed, pairing-free signatures based on ECDSA can reduce the computation time by 95% compared to pairing-based ones. Despite this, the main disadvantage of their research lies in using part of the private key in the algorithm. D_A is part of the private key of user A, so D_A is secret for user A only, albeit that it is sent out to prove the authenticity of user A's signature. Therefore, the private key of a user is not confidential, which conflicts with the concept of a public-key cryptosystem.

3.2 ID-Based Group Key Agreement Without Pairing (Abhimanyu Kumar & Tripathi, 2016)

In this key agreement protocol, the KGC generates private key $sk_1 \in \mathbb{Z}_p^*$ and public key $P_{pub} = sk_1 \bullet P$, where P is a base point on the elliptic curve. Random number r_A generated by the KGC is used to compute the private key for user A. On the user side, the public key for user A is the user's ID (e.g., name, phone number, etc.). The private key for user A is (s_A, D_A); D_A and s_A are respectively computed by the KGC as $D_A = r_A \bullet P$ and $s_A = (r_A + sk_1 \bullet h_A) \bmod p$, where $h_A = HF_2(ID_A)$ is the user's ID hashed using hash function HF_2: $\{0,1\}^* \longrightarrow \{0,1\}^k$, for which k is the bit length of prime p. Similarly, the KGC randomly produces r_B to generate $D_B = r_B \bullet P$ and $s_B = (r_B + sk_1 \bullet h_B) \bmod p$, where $h_B = HF_2(ID_B)$ for user B.

Session private key $x_A \in \mathbb{Z}_p^*$ is randomly chosen for user A and $T_A = x_A \bullet P$ is computed. T_A, D_A is sent on behalf of user A to user B and user A receives T_B, D_B from user B, where $T_B = x_B \bullet P$. Following this, K_A^B is computed for user A as follows:

$$\begin{aligned} K_A^B &= (s_A T_B + x_A (D_B + HF_2(ID_B) \bullet P_{pub})) \\ &= ((r_A + sk_1 \bullet HF_2(ID_A)) \bullet x_B P + x_A (r_B \bullet P + HF_2(ID_B)\, sk_1 \bullet P)) \\ &= ((r_A P + sk_1 \bullet P \bullet HF_2(ID_A))\, x_B + x_A \bullet P\, (r_B + HF_2(ID_B) \bullet sk_1)) \\ &= ((D_A + HF_2(ID_A) \bullet P_{pub})\, x_B + T_A s_B) \\ &= (s_B T_A + x_B (D_A + HF_2(ID_A) \bullet P_{pub})). \end{aligned}$$

Similarly, K_B^A is computed for user B by applying

$$K_B^A = (s_B T_A + x_B (D_A + HF_2(ID_A) \bullet P_{pub})).$$

As a result, the agreed key for messages between users A and B is $K_A^B = K_B^A$.

The disadvantage of the approach in (Abhimanyu Kumar & Tripathi, 2016) is the same as that in (Jin et al., 2010); i.e., part of user *j*'s private key D_j must be sent to proceed with the key agreement algorithm. Thus, the private key is no longer secret, and that violates the concept of a public-key cryptosystem. Moreover, this research is limited to users that belong to the same KGC, while in real-life situations, communication is usually between users from different KGCs.

CHAPTER 4

THE PROPOSED PAIRING-FREE ID-BASED CRYPTOSYSTEM

The proposed ID-based cryptosystem is based on ECC and consists of encryption, digital signature, and key exchange schemes that make use of the same initial definition and key extraction parts. To demonstrate how this cryptosystem works, let us assume that there are three parties: users A, B, and C. Users A and B belong to KGC_1, while user C belongs to KGC_2. Their public and private keys are defined as follows:

For the rest of the thesis, we define $\mathbb{G}$ as the cyclic additive group of points on an elliptic curve over finite field $E/\mathbb{F}_p$.

4.1 System parameter definition

KGC_1 defines base point $P_1 \in \mathbb{G}$ on the elliptic curve in which the order is a large value n and $nP_1 = O$, a hash function HF_3: $\{0,1\}^* \rightarrow \mathbb{Z}_n^*$, and master key $sk_1 \in [1, n - 1]$. Subsequently, KGC_1 computes its public key $Ppub_1 = (sk_1 \cdot P_1)$ and broadcasts $< \mathbb{G}, P_1, Ppub_1, HF_3 >$ as the system parameters of the KGC_1 domain. In the same way, KGC_2 defines base point $P_2 \in \mathbb{G}$, master key $sk_2 \in [1, n - 1]$, and hash function HF_3: $\{0,1\}^* \rightarrow \mathbb{Z}_n^*$ and computes the corresponding $Ppub_2 = (sk_2 \cdot P_2)$. KGC_2 keeps sk_2 secret and broadcasts $< \mathbb{G}, P_2, Ppub_2, HF_3 >$ as the system parameters of the KGC_2 domain.

4.2 Key extraction

In the key extraction phase, the users' public and private keys and some parameters used in the cryptosystem are defined. c_1 and c_2 are strings that are concatenated with the user's ID to generate the private key, which offers better security than the other algorithms. The procedure for defining the key is as follows.

4.2.1 Key extraction for user A

KGC_1 chooses random number $r_A \in [1, n - 1]$ to compute private key s_A for user A as

$$s_A = (sk_1 \cdot HF_3(ID_A) + r_A \cdot HF_3(ID_A \| c_1)) \bmod n.$$

KGC_1 then computes $R_A = r_A \cdot P_1$ and sends s_A, R_A to user A through a secure channel. Upon receiving the message, s_A is kept secret for user A and string ($ID_A.R_A$) is announced as his/her public key. Note that R_A is composed of coordinate (x,y).

4.2.2 Key extraction for user B

KGC_1 chooses random number $r_B \in [1, n - 1]$ to compute private key s_B for user B as

$$s_B = (sk_1 \cdot HF_3(ID_B) + r_B \cdot HF_3(ID_B \| c_1)) \bmod n.$$

KGC_1 then computes $R_B = r_B \cdot P_1$ and sends s_B, R_B to user B securely. Upon receiving the message, s_B is kept secret for user B and broadcast string ($ID_B.R_B$) as his/her public key.

The above key definition has two advantages. First, it can prevent situations where two or more users cooperate to determine the KGC's master key. Second, if a user's private key is compromised, the KGC can easily generate a new secret key for that user. For example, if s_A is compromised, the KGC can randomly generate a new r_A to recompute the new private key.

4.2.3 Key extraction for user C

The private key of user C, who is a member of KGC_2, can be computed as $s_C = (sk_2 \cdot HF_3(ID_C) + r_C \cdot HF_3(ID_C \| c_2)) \bmod n$. Thus, the corresponding public key is $(ID_C.R_C)$, where $R_c = r_c \cdot P_2$.

The pairing-free IBE, digital signature, and key exchange algorithms based on the above definition are introduced in the following sections.

4.3 The encryption and decryption scheme

Using the system parameters in the key extraction phase, encryption and decryption in the pairing-free ID-based key system can be achieved as follows:

Let HF_4: $\{ (x,y) \} \longrightarrow (0,1)^m$.

4.3.1 Encryption

When user B wants to send a secure message to user A, the encrypted message is computed by using user A's public key as follows. For user B, $x_B \in \mathbb{Z}_n^*$ is randomly generated and

$EncM = M \oplus HF_4((Ppub_1 \bullet HF_3(ID_A) + R_A \bullet HF_3(ID_A \| c_1)) \bullet x_B)$

is computed. Subsequently, encrypted message $((x_B \cdot P_1), EncM)$ is sent to user A.

4.3.2 Decryption

Message *EncM* is decrypted for user A by computing

$DecM = EncM \oplus HF_4(s_A \bullet (x_B \cdot P_1))$.

4.3.3 Proof of the correctness of the scheme

The proof is as follows:

$DecM = EncM \oplus HF_4(s_A \bullet (x_B \cdot P_1))$

$= EncM \oplus HF_4\big((sk_1 \bullet HF_3(ID_A) + r_A \bullet HF_3(ID_A \| c_1)) \bullet (x_B \bullet P_1)\big)$

$= EncM \oplus HF_4\big((sk_1 P_1 \bullet HF_3(ID_A) + r_A P_1 \bullet HF_3(ID_A \| c_1)) \bullet x_B\big)$

$= EncM \oplus HF_4\big((Ppub_1 \bullet HF_3(ID_A) + R_A \bullet HF_3(ID_A \| c_1)) \bullet x_B\big)$

$= M$.

4.4 The ID-based digital signature scheme

Our ID-based digital signature algorithm is based on ECDSA, the details of which can be found in (American National Standards Institute, 1998). In the algorithm, let HF_5 be cryptographic hash function SHA-1 giving a 160-bit integer value as the output. When user A wants to send a message together with a signature to user B, signature pair (r,z) is generated for message M by computing the following steps:

1. Randomly generate integer $k \in [1, n - 1]$ for user A, and then compute point $(x,y) = k \cdot P_1$.
2. Compute the first portion of signature $r = x \bmod n$, for $r \neq 0$.
3. Hash message M using $e = HF_5(M)$.
4. Compute the second portion of signature $z = k^{-1}(e+s_A r) \bmod n$, for $z \neq 0$.
5. Send (r,z) as the signature of message M to user B.

When user B receives (r,z) from user A, the following steps are executed to verify the message:

1.Compute $e = HF_5(M)$

2. Compute $Q = Ppub_1 \cdot HF_3(ID_A) + R_A \cdot HF_3(ID_A \| c_1)$

$= sk_1 \cdot P_1 \cdot HF_3(ID_A) + r_A \cdot P_1 \cdot HF_3(ID_A \| c_1)$

$= (sk_1 \cdot HF_3(ID_A) + r_A \cdot HF_3(ID_A \| c_1)) \cdot P_1$

$= s_A \cdot P_1$

3. Compute $w = z^{-1} \bmod n$
4. Compute $u_1 = ew$ and $u_2 = rw$
5. Compute point $(x_1,y_1) = u_1 \cdot P_1 + u_2 \cdot Q$
6. Compute $v = x_1 \bmod n$
7. Accept the signature (r,z) from user A if and only if $v = r$

In ECDSA, it has been proved that if Q is the corresponding public key of the private key used in the signing process, then $v = r$. As we have followed the ECDSA process and because $Q = s_A P_1$, i.e., Q is equivalent to user A's public key in ECDSA, we can conclude that the proposed algorithm is valid.

4.5 The key exchange scheme

For practical implementation, the proposed algorithm is designed to implement key exchange between parties from different KGCs. In the following example, we demonstrate key exchange between user A in KGC_1 and user C in KGC_2.

The process begins with parameter generation and exchange between the parties.

1. $x_A \in [1, n - 1]$ is randomly chosen for user A and then $(x_A \cdot P_2)$ is sent to user C.
2. $x_C \in [1, n - 1]$ is chosen for user C and then $(x_C \cdot P_1)$ is sent to user A, after which both systems cooperate to produce a shared key for the users.
3. K_A^C is computed for user A:

$$K_A^C = s_A (x_C \cdot P_1) + x_A \left(R_C \cdot HF_3(ID_C \| c_2) + Ppub_2 \cdot HF_3(ID_C) \right).$$

4. K_C^A is computed for user C:

$$K_C^A = (Ppub_1 \cdot HF_3(ID_A) + R_A \cdot HF_3(ID_A \| c_1)) x_C + (x_A \cdot P_2) s_C,$$

after which the shared key between users A and C becomes $K_A^C = K_C^A$.

The following proof demonstrates that K_A^C and K_C^A are equal. Recall that

$$s_A = (sk_1 \cdot HF_3(ID_A) + r_A \cdot HF_3(ID_A \| c_1)) \bmod n,$$

$$s_C = (sk_2 \cdot HF_3(ID_C) + r_C \cdot HF_3(ID_C \| c_2)) \bmod n,$$

$$R_A = r_A \cdot P_1, R_C = r_C \cdot P_2,$$

$$Ppub_1 = (sk_1 \cdot P_1), Ppub_2 = (sk_2 \cdot P_2).$$

$$\begin{aligned} K_A^C &= s_A(x_C \cdot P_1) + x_A (R_C \cdot HF_3(ID_C \| c_2) + Ppub_2 \cdot HF_3(ID_C)) \\ &= (sk_1 \cdot HF_3(ID_A) + r_A \cdot HF_3(ID_A \| c_1)) \cdot (x_C \cdot P_1) \\ &\quad + x_A ((r_C \cdot P_2) \cdot HF_3(ID_C \| c_2) + (sk_2 \cdot P_2) \cdot HF_3(ID_C)) \\ &= (sk_1 \cdot P_1 \cdot HF_3(ID_A) + r_A \cdot P_1 \cdot HF_3(ID_A \| c_1)) \cdot x_C \\ &\quad + x_A \cdot P_2 (r_C \cdot HF_3(ID_C \| c_2) + sk_2 \cdot HF_3(ID_C)) \\ &= (Ppub_1 \cdot HF_3(ID_A) + R_A \cdot HF_3(ID_A \| c_1)) x_C + (x_A \cdot P_2) s_C \\ &= K_C^A. \end{aligned}$$

As illustrated above, the proposed encryption algorithm, digital signature, and key exchange processes make use of the same key definition and system parameters. The other main advantage of the key definition is that if the current user's private key is compromised, we can change the private key easily without affecting the user's ID. However, since R_A is not authenticated, there is the chance that an attacker will try to change the value of R_A to carry out attacks such as man-in-the-middle or using a fake signature. Fortunately, our proposed system can endure such attacks, as discussed in the next chapter.

CHAPTER 5

SECURITY AND PERFORMANCE ANALYSIS OF THE PROPOSED ALGORITHMS

In this chapter, we discuss the security of our proposed cryptosystem. The security analysis was conducted using a man-in-the-middle attack on the encryption and key exchange schemes. Moreover, problems with private key recovery and fake signatures for the digital signature scheme are discussed. In the last section, we analyze and compare the performance between our proposed pairing-free scheme with some well-known pairing-based ones.

5.1 Security analysis of the encryption algorithm

One major issue for an encryption system is vulnerability to a man-in-the-middle attack. In brief, this is a situation whereby an eavesdropper tries to intercept messages between users and sometimes modifies them without being detected. In this analysis, we suppose that user X, who is a member of KGC_1, impersonates user A and modifies user A's public key $ID_A.R_A$ to fraudulently created public key $ID_A.R_X$.

Let us define user X's private keys as

$$s_X = (sk_1 \cdot HF_3(ID_X) + r_X \cdot HF_3(ID_X \| c_1)) \bmod n$$

The fake public key of user $A = ID_A.R_X$, where $R_X = r_X \cdot P_1$

Later, user B wants to send a secret message to user A. However, the message from user B is encrypted with the fraudulent public key, $ID_A.R_X$. Thus, message M is encrypted by computing

Random $x_B \in \mathbb{Z}_n^*$,

$$EncM = M \oplus HF_4(\ (Ppub_1 \bullet HF_3(ID_A) + R_X \bullet HF_3(ID_A \| c_1)) \bullet x_B\ ).$$

After that, user B sends $((x_B \cdot P_1), EncM)$ to user A.

User X eavesdrops on message $((x_B \cdot P_1), EncM)$ in the network system and he/she tries to recover the message M by using his/her private key s_X. However, user X fails to recover message M from $EncM$ as we demonstrate below:

$$\begin{aligned} DecM &= EncM \oplus HF_4(s_X \bullet (x_B \cdot P_1)) \\ &= EncM \oplus HF_4(s_X \bullet (x_B \cdot P_1)) \\ &= EncM \oplus HF_4((sk_1 \bullet HF_3(ID_X) + r_X \bullet HF_3(ID_X \| c_1)) \bullet (x_B \bullet P_1)) \\ &= EncM \oplus HF_4((sk_1 P_1 \bullet HF_3(ID_X) + r_X \bullet P_1 \bullet HF_3(ID_X \| c_1)) \bullet x_B) \\ &= M \oplus HF_4((Ppub_1 \bullet HF_3(ID_A) + R_X \bullet HF_3(ID_A \| c_1)) \bullet x_B) \\ &\quad \oplus HF_4((Ppub_1 \bullet HF_3(ID_X) + R_X \bullet HF_3(ID_X \| c_1)) \bullet x_B) \\ &\neq M\ . \end{aligned}$$

Thus, the probability that user X can carry out a man-in-the-middle attack is

$$prob\{HF_3(ID_A) = HF_3(ID_X)\} \times prob\{HF_3(ID_A \| c_1) = HF_3(ID_X \| c_1)\}.$$

One interesting question is how to tell whether public R_A is genuine. We can test the genuineness of R_A by using the protocol; i.e., a message with user A's public key can be encrypted and sent to user A. If the responding message from user A has the correct signature, we can tell that public R_A is genuine.

5.2 Security analysis of the digital signature scheme

The main problem with digital signatures is in situations where user X impersonates user A and sends a message with a fake digital signature to a receiver. If the receiver verifies the fake signature by using the fake public key of user A, i.e. $ID_A.R_X$, then the attack is successful. However, our proposed digital scheme can resist such attack as described below.

User X creates a fake digital signature for message M as follows:

1. Randomly create integer $k \in [1, n - 1]$, and then compute $(x,y) = k \cdot P_1$.
2. Compute first signature $r = x \bmod n$, for $r \neq 0$
3. Hash message M by using $e = HF_5(M)$

4. Compute $z = k^{-1}(e+s_X r) \bmod n$, for $z \neq 0$

5. Send (r,z) to user B

Upon receiving the fake signature, user signature (r,z) is verified for user B by using the fake public key $(ID_A.R_X)$ of user A. The signature verification process is shown below.

1. Compute $e = HF_5(M)$

2. Compute $Q = ((Ppub_1 \cdot HF_3(ID_A)) + (R_X \cdot HF_3(ID_A \| c_1)))$

3. Compute $w = z^{-1} \bmod n$

4. Compute $u_1 = ew$ and $u_2 = rw$

5. Compute point $(x_1,y_1) = u_1 \cdot P_1 + u_2 \cdot Q$

6. Compute $v = x_1 \bmod n$

7. Compare v with r.

According to ECDSA, if $r = v$, then the verification is successful. As we showed in chapter 4, $r = v$ if $Q = s_X \cdot P_1$. However, the above calculation shows that $Q = ((Ppub_1 \cdot HF_3(ID_A))+(R_X \cdot HF_3(ID_A \| c_1))) \neq s_X \cdot P_1$. Therefore, (r,z) is rejected on behalf of user B as a fake signature. Furthermore, the probability of an attack being successful is

$$prob\{HF_3(ID_A) = HF_3(ID_X)\} \times prob\{HF_3(ID_A \| c_1) = HF_3(ID_X \| c_1)\}.$$

Another benefit that our proposed protocol inherits from ECDSA is that ECDSA is secure and can prevent private key recovery from signature (r,z). As the calculation of pair (r,z) in our scheme follows ECDSA, the scheme can endure a key recovery attack.

5.3 Security analysis of the key exchange scheme

A man-in-the-middle-attack is a major concern for a key exchange algorithm. In the key exchange process, when the parameters for two users (e.g., users A and B) are exchange when proceeding with the key exchange process, user X (the attacker) can impersonate user B and take part in the key exchange process with user A. The system interprets that an agreed key between users A and B has been created, but it is really between users A and X. In the same way, user X impersonates user A and takes

part in the key exchange process with user B. Hence, user X can eavesdrop on or edit the communication between users A and B.

In this section, the security for two circumstances of using the key exchange algorithm is analyzed: two parties belonging to different KGCs and two in the same KGC.

5.3.1 Two parties on different KGCs

Let us assume that a shared key between user A belonging to KGC_1 and user C belonging to KGC_2 needs to be established. Meanwhile, the attacker (user X) belongs to KGC_1. In the key exchange process, the algorithm described in chapter 4.3 is executed to establish a shared key for user A. Afterward, $(x_A \cdot P_2)$ on behalf of user A is sent out into the network. At that time, user X intercepts the message and sends $(x_X \cdot P_1)$ to user A. Upon receiving the message, the fake public key of user C, $(ID_C.R_X)$, is used by the system to compute a shared key for user A as follows:

$$\begin{aligned} K_A^X &= s_A \cdot (x_X \cdot P_1) + x_A \left(R_X \cdot HF_3(ID_C \,||\, c_2) + Ppub_2 \cdot HF_3(ID_C)\right) \\ &= (sk_1 \cdot HF_3(ID_A) + r_A \cdot HF_3(ID_A \,||\, c_1)) \cdot (x_X \cdot P_1) \\ &\quad + x_A \left((r_X \cdot P_2) \cdot HF_3(ID_C \,||\, c_2) + (sk_2 \cdot P_2) \cdot HF_3(ID_C)\right) \\ &= (sk_1 \cdot P_1 \cdot HF_3(ID_A) + r_A \cdot P_1 \cdot HF_3(ID_A \,||\, c_1)) \cdot x_X \\ &\quad + x_A \cdot P_2 \left(r_X \cdot HF_3(ID_C \,||\, c_2) + sk_2 \cdot HF_3(ID_C)\right). \end{aligned}$$

Meanwhile, $(x_A \cdot P_2)$ is used by user X to compute

$$\begin{aligned} K_X^A &= (Ppub_1 \cdot HF_3(ID_A) + R_A \cdot HF_3(ID_A \,||\, c_1)) \, x_X + (x_A \cdot P_2) \, s_X \\ &= (sk_1 \cdot P_1 \cdot HF_3(ID_A) + r_A \cdot P_1 \cdot HF_3(ID_A \,||\, c_1)) \cdot x_X \\ &\quad + x_A \cdot P_2 \, (r_X \cdot HF_3(ID_X \,||\, c_2) + sk_2 \cdot HF_3(ID_X)). \end{aligned}$$

Consequently, we can see that $K_A^X \neq K_X^A$, and thus user X fails to establish a shared key with user A. Moreover, the probability that the attack is successful is

$$prob\{HF_3(ID_C \,||\, c_2) = HF_3(ID_X \,||\, c_2)\} \times prob\{HF_3(ID_C) = HF_3(ID_X)\}.$$

In the same way, user X impersonates user A and tries to establish a shared key with user C. In this process, $(x_C \cdot P_1)$ is sent to the network on behalf of user C,

and user X intercepts the message and sends $(x_X \cdot P_2)$ back to user C. Next, the fake public key of user A $(ID_A.R_X)$ is used to compute a shared key with user C as follows:

$$K_C^X = (Ppub_1 \cdot HF_3(ID_A) + R_X \cdot HF_3(ID_A \| c_1)) x_C + (x_X \cdot P_2) \cdot s_C$$
$$= (sk_1 \cdot P_1 \cdot HF_3(ID_A) + r_X \cdot P_1 \cdot HF_3(ID_A \| c_1)) \cdot x_C$$
$$+ x_X \cdot P_2 (r_C \cdot HF_3(ID_C \| c_2) + sk_2 \cdot HF_3(ID_C)).$$

Meanwhile, user X computes K_X^C using $(x_C \cdot P_1)$:

$$K_X^C = s_X (x_C \cdot P_1) + x_X (R_C \cdot HF_3(ID_C \| c_2) + Ppub_2 \cdot HF_3(ID_C))$$
$$= (sk_1 \cdot HF_3(ID_X) + r_X \cdot HF_3(ID_X \| c_1)) \cdot (x_C \cdot P_1)$$
$$+ x_X ((r_C \cdot P_2) \cdot HF_3(ID_C \| c_2) + (sk_2 \cdot P_2) \cdot HF_3(ID_C))$$
$$= (sk_1 \cdot P_1 \cdot HF_3(ID_X) + r_X \cdot P_1 \cdot HF_3(ID_X \| c_1)) \cdot x_C$$
$$+ x_X \cdot P_2 (r_C \cdot HF_3(ID_C \| c_2) + sk_2 \cdot HF_3(ID_C)).$$

Hence, we can see that $K_C^X \neq K_X^C$, and thus user X fails to establish a shared key with user C. Moreover, the probability that user X can establish a shared key with user C is

$$prob\{HF_3(ID_A) = HF_3(ID_X)\} \times prob\{HF_3(ID_A \| c_1) = HF_3(ID_X \| c_1)\}.$$

Even if user X eavesdrops on the communication between users A and C, he/she cannot recover or edit the message. We can conclude that the proposed scheme can prevent a man-in-the-middle attack in cases where the key exchange parties are members of different KGCs.

5.3.2 Two parties on the same KGC

Let us assume that users A, B, and X belong to KGC_1. In the key exchange process, $(x_A \cdot P_1)$ is sent out into the network on behalf of user B, and then user X intercepts the message and sends $(x_X \cdot P_1)$ to user A. The fake public key of user B $(ID_B.R_X)$ is used on behalf of user A to compute a shared key as follows:

$$K_A^X = s_A \cdot (x_X \cdot P_1) + x_A (R_X \cdot HF_3 (ID_B \| c_1) + Ppub \cdot HF_3(ID_B))$$
$$= (sk_1 \cdot HF_3(ID_A) + r_A \cdot HF_3(ID_A \| c_1)) \cdot (x_X \cdot P_1)$$
$$+ x_A ((r_X \cdot P_1) \cdot HF_3(ID_B \| c_1) + (sk_1 \cdot P_1) \cdot HF_3(ID_B))$$
$$= (sk_1 \cdot P_1 \cdot HF_3(ID_A) + r_A \cdot P_1 \cdot HF_3(ID_A \| c_1)) \cdot x_X$$
$$+ x_A \cdot P_1 (r_X \cdot HF_3(ID_B \| c_1) + sk_1 \cdot HF_3(ID_B)).$$

Meanwhile, user X uses $(x_A \cdot P_1)$ to compute a shared key:

$$K_X^A = (Ppub \cdot HF_3 (ID_A) + R_A \cdot HF_3 (ID_A \| c_1)) x_X + (x_A \cdot P_1) \cdot s_X$$
$$= (sk_1 \cdot P_1 \cdot HF_3(ID_A) + r_A \cdot P_1 \cdot HF_3(ID_A \| c_1)) \cdot x_X$$

$$+ x_A \cdot P_1 \, (r_X \cdot HF_3(ID_X \| c_1) + sk_1 \cdot HF_3(ID_X)).$$

According to the above computation, we can see that $K_A^X \neq K_X^A$. Thus, user X fails to establish a shared key with user A. Furthermore, the probability that the attack is successful is

$$prob\{HF_3(ID_B \| c_1) = HF_3(ID_X \| c_1)\} \times prob\{HF_3(ID_B) = HF_3(ID_X)\}.$$

In the same way, user X impersonates user A and tries to establish a shared key with user B. ($x_B \cdot P_1$) is sent out into the network on behalf of user B, and then user X intercepts the message and sends ($x_X \cdot P_1$) to user B. The fake public key of user A ($ID_A.R_X$) is then used to compute the shared key with user B as

$$\begin{aligned} K_B^X &= (Ppub \cdot HF_3\,(ID_A) + R_X \cdot HF_3\,(ID_A \| c_1))\, x_B + (x_X \cdot P_1) s_B \\ &= (sk_1 \cdot P_1 \cdot HF_3(ID_A) + r_X \cdot P_1 \cdot HF_3(ID_A \| c_1)) \cdot x_B \\ &\quad + x_X \cdot P_1\, (r_B \cdot HF_3(ID_B \| c_1) + sk_1 \cdot HF_3(ID_B)). \end{aligned}$$

Meanwhile, user X uses ($x_B \cdot P_1$) to compute K_X^B as follows:

$$\begin{aligned} K_X^B &= s_X\ (x_B \cdot P_1) + x_X\ (\ R_B \cdot\ HF_3(ID_B \| c_1) + Ppub_1 \cdot HF_3\ (ID_B)) \\ &= (sk_1 \cdot HF_3(ID_X) + r_X \cdot HF_3(ID_X \| c_1)) \cdot x_B \cdot P_1 \\ &\quad + x_X\ (\ (r_B \cdot P_2) \cdot HF_3(ID_B \| c_1) + (sk_1 \cdot\ P_1) \cdot HF_3(ID_B)) \\ &= (sk_1 \cdot P_1 \cdot HF_3(ID_X) + r_X \cdot P_1 \cdot HF_3(ID_X \| c_1)) \cdot x_B \\ &\quad + x_X \cdot P_1\, (r_B \cdot HF_3(ID_B \| c_1) + sk_1 \cdot HF_3(ID_B)). \end{aligned}$$

Hence, we can see that $K_B^X \neq K_X^B$, and so user X fails to establish a shared key with user B. Moreover, the probability that the attack is successful is

$$prob\{HF_3(ID_A) = HF_3(ID_X)\} \times prob\ \{HF_3(ID_A \| c_1) = HF_3(ID_X \| c_1)\}.$$

From the findings of the above analysis, it can be seen that our proposed algorithm can prevent man-in-the-middle attacks if users are members of the same KGC or otherwise.

5.4 Performance analysis of the proposed cryptosystem.

Table 5.1 reports the results of a performance comparison between our proposed pairing-free scheme with well-known pairing-based ones.

Table 5.1 Comparison of well-known Pairing-Based schemes and the proposed Pairing-Free scheme.

Algorithm	Scheme	Number of Pairings $\hat{e}(U,V)$	Number of Scalar Multiplications on the Elliptic Curve	Number of Integer Multiplications	Number of $\hat{e}(U,V)\cdot\hat{e}(U,V)$	Number of Pairings $\hat{e}(U,V)^a$
Key Extract	Pairing-based (Boneh & Franklin, 2003)	-	1; i.e. $S_{ID} = sk_1 \bullet Q_{ID}$	-	-	-
	Pairing-free	-	1; i.e. $R_j = r_j \cdot P_i$	2; i.e. $sk_i \cdot HF_3(ID_j)$, $r_j \cdot HF_3(ID_j \| c_i)$	-	-
Encryption	Pairing-based (Boneh & Franklin, 2003)	-	1; i.e. $r \bullet P$	-	-	1; i.e. $\hat{e}\left(P_{pub}, Q_{ID}\right)^r$
	Pairing-free	-	4; i.e. $Ppub_i \bullet HF_3(ID_A)$, $(R_A \bullet HF_3(ID_A \| c_i)) \bullet x_B$, $x_B \cdot P_i$	-	-	-
Decryption	Pairing-based (Boneh & Franklin, 2003)	1; i.e. $\hat{e}(rP, S_{ID})$	-	-	-	-
	Pairing-free	-	1; i.e. $s_A \bullet (x_B \cdot P_1)$	-	-	-
Digital Signature Signing Process	Pairing-based (Paterson, 2002)	-	4; i.e. $R = k \bullet P$, $Z = k^{-1}(H_3(m) \bullet P + H_4(R) \bullet S_{ID})$	-	-	-
	Pairing-free	-	1; i.e. $k \cdot P_i$	2; i.e. $k^{-1}(e+s_j r)$	-	-

Algorithm	Scheme	Number of Pairings $\hat{e}(U, V)$	Number of Scalar Multiplications on the Elliptic Curve	Number of Integer Multiplications	Number of $\hat{e}(U, V)\cdot \hat{e}(U, V)$	Number of Pairings $\hat{e}(U, V)^a$
Digital Signature Verification Process	Pairing-based (Paterson, 2002)	1; i.e. $\hat{e}(R,Z)$	-	-	1; i.e. $\hat{e}(P,P)^{H_3(m)} \cdot \hat{e}(sk_1P, Q_{ID})^{H_4(R)}$	2; i.e. $\hat{e}(P,P)^{H_3(m)}$ $\hat{e}(sk_1P, Q_{ID})^{H_4(R)}$
	Pairing-free	-	4; i.e. $Ppub_1 \cdot HF_3(ID_A)$, $R_A \cdot HF_3(ID_A \| c_1)$, $u_1 \cdot P_1$, $u_2 \cdot Q$	2; i.e. $u_1 = ew$ $u_2 = rw$	-	-
Key Exchange	Pairing-based (Smart, 2002)	2; i.e. $\hat{e}(aQ_B, P_{KGS})$ $\hat{e}(S_A, T_B)$	2; i.e. aQ_B , $T_A = aP$	-	1; i.e. $\hat{e}(aQ_B, P_{KGS}) \cdot \hat{e}(S_A, T_B)$	-
	Pairing-free	-	5; i.e. $s_A (x_C \cdot P_1)$, $x_A (R_C \cdot HF_3(ID_C \| c_2) + Ppub_2 \cdot HF_3(ID_C))$	-	-	-

Table 5.2 provides the results of a performance comparison between the proposed pairing-free scheme with other well-known pairing-free ones.

Table 5.2 Comparison of well-known Pairing-Free schemes and the proposed Pairing-Free scheme.

Algorithm	Scheme	Number of Scalar Multiplications on the Elliptic Curve	Number of Integer Multiplications	Sending out the User's Private Key	Key Exchange Between Users Belonging to Different KGCs
Key Extract	(Jin et al., 2010), (Abhimanyu Kumar & Tripathi, 2016)	1; i.e. $D_A = r_A \cdot P$	1; i.e. $s_A = (r_A + h \cdot sk_1)$	-	-
	The proposed algorithm	1; i.e. $R_A = r_A \cdot P_1$	2; i.e. $s_A = sk_1 \cdot HF_3(ID_A) + r_A \cdot HF_3(ID_A \| c_1)$	-	-
Digital Signature Signing Process	(Jin et al., 2010)	1; i.e. $(x_1,y_1) = k \cdot P_1$	2; i.e. $S = k^{-1}(e+dr)$	yes	-
	The proposed algorithm	1; i.e. $(x,y) = k \cdot P_1$	2; i.e. $z = k^{-1}(e+s_A r)$	no	-

Algorithm	Scheme	Number of Scalar Multiplications on the Elliptic Curve	Number of Integer Multiplications	Sending out the User's Private Key	Key Exchange Between Users Belonging to Different KGCs
Digital Signature Verification Process	(Jin et al., 2010)	3; i.e. $Q = D_A + h \cdot P_{pub}$ $u_1 \cdot P_1$ $u_2 \cdot Q$	2; i.e. $u_1 = ew$ $u_2 = rw$	-	-
	The proposed algorithm	4; i.e. $Q = Ppub_1 \cdot HF_3(ID_A) + R_A \cdot HF_3(ID_A \| c_1)$ $u_1 \cdot P_1$ $u_2 \cdot Q$	2; i.e. $u_1 = ew$ $u_2 = rw$	-	-
Key Exchange	(Abhimanyu Kumar & Tripathi, 2016)	4; i.e. $T_A = x_A \bullet P$ $K_A^B = (s_A T_B + x_A (D_B + HF_2(ID_B) \bullet P_{pub}))$	-	yes	no
	The proposed algorithm	5; i.e. $K_A^C = s_A(x_C \cdot P_1) + x_A (R_C \cdot HF_3(ID_C \| c_2) + Ppub_2 \cdot HF_3(ID_C))$	-	no	yes

CHAPTER 6

CONCLUSIONS

A pairing-free IBC using ECC and consisting of IBE, digital signature, and key exchange schemes was presented. All of the schemes use the same public and private key definitions, which makes the implementation of the system easy. The main advantage of the proposed key definition system is that if the user's private key is compromised, the KGC can easily generate a new one. As for the ID-based key exchange scheme, the proposed system can cope with situations where the communicating parties are on different KGCs. This is useful for mobile network computing in real scenarios. Proof of its correctness and a security analysis were provided, and the durability of the proposed system to several types of attacks (man-in-the-middle and intercepting signatures) was established. The proposed pairing-free scheme was compared with some well-known pairing-based and pairing-free ones, the results of which show that our proposed scheme gave a better performance than the pairing-based ones.

BIBLIOGRAPHY

American National Standards Institute. (1998). *American National Standard X9.62– Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm*. Washington, DC, USA: American Bankers Association.

Benhamouda, F., Couteau, G., Pointcheval, D., & Wee, H. (2015). *Implicit Zero-Knowledge Arguments and Applications to the Malicious Setting.* Paper presented at the Advances in Cryptology -- CRYPTO 2015, Berlin, Heidelberg.

Boneh, D., & Franklin, M. K. (2003). Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing, 32(3)*, 286-615.

Cao, X., Kou, W., & Du, X. (2010). A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. *Information Sciences, 180*(15), 2895-2903. doi:https://doi.org/10.1016/j.ins.2010.04.002

Chakraborty, S., Raghuraman, S., & Rangan, C. P. (2016). A pairing-free, one round identity based authenticated key exchange protocol secure against memory-scrapers. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 7(1)*, 1-22.

Gentry, C., & Silverberg, A. (2002). *Hierarchical ID-Based Cryptography.* Paper presented at the Advances in Cryptology — ASIACRYPT 2002, Berlin, Heidelberg.

Hess, F. (2003). *Efficient Identity Based Signature Schemes Based on Pairings.* Paper presented at the Selected Areas in Cryptography, Berlin, Heidelberg.

Hoffstein, J., Pipher, j., & Silverman, J. H. (2008). *An introduction to mathematical cryptography*. Berlin, Heidelberg: Springer.

Hölbl, M., Welzer, T., & Brumen, B. (2012). An improved two-party identity-based authenticated key agreement protocol using pairings. *Journal of Computer and System Sciences, 78*(1), 142-150. doi:https://doi.org/10.1016/j.jcss.2011.01.002

Islam, S. K. H., & Biswas, G. P. (2011). A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *Journal of Systems and Software, 84*(11), 1892-1898. doi:https://doi.org/10.1016/j.jss.2011.06.061

Islam, S. K. H., & Biswas, G. P. (2017). A pairing-free identity-based two-party authenticated key agreement protocol for secure and efficient communication. *Journal of King Saud University - Computer and Information Sciences, 29*(1), 63-73. doi:https://doi.org/10.1016/j.jksuci.2015.01.004

Jin, H., Debiao, H., & Jianhua, C. (2010, 6-7 March 2010). *An Identity Based Digital Signature from ECDSA.* Paper presented at the 2010 Second International Workshop on Education Technology and Computer Science.

Joux, A. (2000). *A One Round Protocol for Tripartite Diffie–Hellman.* Paper presented at the Algorithmic Number Theory, Berlin, Heidelberg.

Koblitz, A. H., Koblitz, N., & Menezes, A. (2011). Elliptic curve cryptography: The serpentine course of a paradigm shift. *Journal of Number Theory, 131*(5), 781-814. doi:https://doi.org/10.1016/j.jnt.2009.01.006

Kumar, A., & Tripathi, S. (2015). A Pairing Free Anonymous Certificateless Group Key Agreement Protocol for Dynamic Group. *Wireless Personal Communications, 82*(2), 1027-1045. doi:10.1007/s11277-014-2264-3

Kumar, A., & Tripathi, S. (2016). Anonymous ID-based group key agreement protocol without pairing. *International Journal of Network Security, 18(2)*, 263-273.

Kumar, A., Tripathi, S., & Jaiswal, P. (2015, 10-13 Aug. 2015). *Design of efficient ID-based group key agreement protocol suited for Pay-TV application.* Paper presented at the 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

Kumar, M., Katti, C. P., & Saxena, P. (2017). An Identity-based Blind Signature Approach for E-voting System. *International Journal of Modern Education and Computer Science, 9*, 47-54. doi:10.5815/ijmecs.2017.10.06

Lee, H., & Lee, Y. (2005). Identity based authenticated key agreement from parings. *Communications of the Korean Mathematical Society, 20(4)*, 849-859.

Li, H., Dai, Y.-S., & Yang, B. (2011). Identity-Based Cryptography for Cloud Security. *IACR Cryptol. ePrint Arch., 2011*, 169. Retrieved from https://eprint.iacr.org/2011/169.pdf

Liu, L., Cao, Z., Kong, W., & Wang, J. (2017). On bilinear groups of a large composite order. *International Journal of Electronics and Information Engineering, 7(1)*, 1-9.

Malina, L., Hajny, J., & Zeman, V. (2015, 9-11 July 2015). *Usability of pairing-based cryptography on smartphones.* Paper presented at the 2015 38th International Conference on Telecommunications and Signal Processing (TSP).

Ming, Y., & Yuan, H. (2019). Fully secure anonymous identity based broadcast encryption with group of prime order. *International Journal of Network Security, 21(1)*, 7-16.

Naresh, V. S., & Murthy, N. V. E. S. (2015). elliptic curve based dynamic contributory group key agreement protocol for secure group communication over ad-hoc networks. *International Journal of Network Security, 17(5)*, 588-596.

Nathani, S., Tripathi, B., & Khatoon, S. (2019). A dynamic ID based authenticated group key agreement protocol from pairing. *International Journal of Network Security, 21(4)*, 582–591.

Paterson, K. G. (2002). ID-based Signature from Pairings on Elliptic Curves. *Electronics Letters, 38(18)*, 1025-1026. doi:10.1049/el:20020682

Roy, S., & Khatwani, C. (2017). Cryptanalysis and Improvement of ECC Based Authentication and Key Exchanging Protocols. *Cryptography, 1(1)*, 9. doi:10.3390/cryptography1010009

Sakai, R., & Kasahara, M. (2003). ID based Cryptosystems with Pairing on Elliptic Curve. *IACR Cryptology ePrint Archive, 2003*. Retrieved from http://eprint.iacr.org/2003/054

Shamir, A. (1985). *Identity-Based Cryptosystems and Signature Schemes.* Paper presented at the Advances in Cryptology, Berlin, Heidelberg.

Smart, N. P. (2002). An Identity-based authenticated key agreement protocol based on weil pairing. *Electronics Letters, 38(13)*, 630-632. doi:10.1049/el:20020387

Stalling, W. (2013). *Cryptography and network security principles and practice (6th edn.)*. Hoboken, NJ, USA: Pearson.

Xiaozhuo, G., Taizhong, X., Weihua, Z., & Yongming, W. (2014, 20-22 Aug. 2014). *A Pairing-Free Certificateless Authenticated Group Key Agreement Protocol.* Paper presented at the 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and

Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS).

Youngblood, C. (2005). An introduction to identity-based cryptography: CSEP 590TU. Retrieved from https://courses.cs.washington.edu/courses/csep590/06wi/finalprojects/youngblood_csep590tu_final_paper.pdf

Zhandry, M. (2012). *Secure Identity-Based Encryption in the Quantum Random Oracle Model.* Paper presented at the Advances in Cryptology – CRYPTO 2012, Berlin, Heidelberg.

Zhu, R. W., Yang, G., & Wong, D. S. (2005). *An Efficient Identity-Based Key Exchange Protocol with KGS Forward Secrecy for Low-Power Devices.* Paper presented at the Internet and Network Economics, Berlin, Heidelberg.

ปญฺญา สมา อาภา
สถาบันบัณฑิตพัฒนบริหารศาสตร์

BIOGRAPHY

Name-Surname Poonsuk Ponpurmpoon

Academic Background M.Sc. (Computer Science and Information Systems), National Institute of Development Administration (NIDA), Thailand.
B.A. (Information Studies), Ramkhamhaeng University, Thailand.