GENERALIZED ID-BASED ELGAMAL SIGNATURES AND EXTENSIONS

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Said Kalkan

July, 2008

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Ali Aydın Selçuk(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ali Doğanaksoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Uluç Saranlı

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

GENERALIZED ID-BASED ELGAMAL SIGNATURES AND EXTENSIONS

Said Kalkan
M.S. in Computer Engineering
Supervisor: Assist. Prof. Dr. Ali Aydın Selçuk
July, 2008

ID-based cryptography helps us to simplify key management process in traditional public key infrastructures. Any public information such as the e-mail address, name, etc., can be used as a public key and this solves the problem of obtaining the public key of a party and checking that its certificate is valid. ID-based cryptography has been a very active area of research in cryptography since bilinear pairings were introduced as a cryptographic tool. There have been many proposals for ID-based signatures recently. In this thesis, we introduce the concept of generalized ID-based ElGamal signatures and show that most of the proposed ID-based signature schemes in the literature are special instances of this generalized scheme. We also investigate ID-based signatures providing additional properties. Signature schemes with message recovery provide the feature that the message is recoverable from the signature and hence does not need to be transmitted separately. Blind signatures provide the feature that a user is able to get a signature without giving the actual message to the signer. Finally, signcryption schemes fulfill the job of a digital signature and encryption in a single step with a lower computational cost.

We generalize the ID-based signatures providing these properties and obtain numerous new signatures which have not been explored before. The generalized ID-based signatures we described provide a unified framework for ID-based ElGamal signatures and extensions. Additionally, some of our blind signatures turn out to be more efficient than the previously proposed schemes.

Keywords: ID-Based Signature, Blindness, Message Recovery, Signcryption.

ÖZET

GENELLEŞTİRİLMİŞ KİMLİK TABANLI ELGAMAL İMZALARI VE EKLENTİLERİ

Said Kalkan
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Assist. Prof. Dr. Ali Aydın Selçuk
Temmuz, 2008

Kimlik tabanlı kriptografi, geleneksel açık anahtar altayapılarındaki anahtar yönetim süreçlerinin basitleştirilmesinde kullanılır. Genele açık her türlü bilgi, örneğin e-posta adresi, ad-soyad açık anahtar olarak kullanılabilir. Bu sayede, bir grubun açık anahtarını elde etme ve bu anahtarın sertifikasının geçerli olup olmadığını kontrol etme süreçleri problem olmaktan çıkar. İkidoğrusal eşlemelerin kriptografide kullanılmaya başlamasından sonra, kimlik tabanlı kriptografi çok aktif bir araştırma alanı olmuştur. Son zamanlarda çok sayıda kimlik tabanlı elektronik imza önerileri sunulmuştur. Bu tezde, genelleştirilmiş kimlik tabanlı ElGamal imza yöntemlerini sunuyoruz ve genelleştirilmiş yöntemin literatürdeki kimlik tabanlı imza yöntemlerinin bir çoğunu kapsadığını gösteriyoruz. Bunun yanında, ekstra özellikler sunan kimlik tabanlı imzaları da inceliyoruz. İmzadan mesajı yeniden inşa etmeye olanak sağlayan imza yöntemlerinde mesajın imzadan ayrı olarak gönderilmesine gerek yoktur. Görmeden imza yöntemi bir kullanıcının imzalatmak istediği mesajı, imzalayacak kişiye vermeden o mesajı imzalatmasına olanak sağlar. İmzala-şifrele yöntemi, elektronik imza ve şifrelemenin tek bir adımda daha az işlem gücü kullanılarak yapılmasını sağlar.

Bu tezde, ekstra özellikler sunan kimlik tabanlı imzalama yöntemlerini genelleştirdik ve birçok daha önce önerilmemiş imza yöntemi elde ettik. Önerdiğimiz genelleştirilmiş kimlik tabanlı imzalar, kimlik tabanlı ElGamal imzaları ve eklentileri için genel bir taslak oluşturuyor. Ek olarak, bazı görmeden imzalama yöntemlerimiz daha önce önerilmiş yöntemlerden daha hızlı çalışıyor.

Anahtar sözcükler: Kimlik Tabanlı İmza, Görmeden İmzalama, İmzadan Mesaj Oluşturma, İmzala-Şifrele Yöntemi.

Acknowledgement

I would like to thank Dr. Ali Aydın Selçuk for offering his valuable time and support as my supervisor. I have learned a lot from his suggestions during this research. I also would like to thank Kamer Kaya and the other members of Security Reading Group (SRG) who helped me during this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1984, Shamir [**?**] introduced the concept of ID-based cryptography to simplify key management procedures in public key infrastructures (PKI). Following Joux's [**?**] discovery on how to utilize bilinear pairings in public key cryptosystems, ID-based cryptography has become one of the most active research areas in cryptography and numerous ID-based encryption, signature and key agreement schemes have been proposed, mostly using bilinear pairings.

In a traditional public key cryptosystem, the user has to obtain a digital certificate issued by a Certificate Authority (CA). A digital certificate contains the public key and identity of the user and proves that the public key in the certificate belongs to the user's identity. To achieve this, the validity of a certificate should be verified by checking the certificate revocation list published by the CA. Usually public key infrastructure is hierarchical and many CAs are involved between the sender and receiver, hence the entire certificate path has to be verified. Therefore obtaining the public key of a party and checking that its certificate is valid is the main problem in traditional public key cryptosystems.

ID-based cryptography helps us to simplify the key management process in traditional PKIs. In ID-based cryptography, any public information such as e-mail address, name, etc., can be used as a public key. Since public keys are derived from publicly known information, their authenticity is established inherently and

there is no need for certificates in ID-based cryptography. The private key for a given public key is generated by a trusted authority (TA) and is sent to the user over a secure channel. Furthermore, public and private keys do not have to be generated at the same time, so Alice can send an encrypted message to Bob even before Bob obtains his private key. After Bob receives the encrypted message, he can ask the TA for his private key corresponding to his identity.

Simplified key distribution and non-requirement for public key directories make ID-based cryptosystems advantageous over traditional PKIs. However, there is an inherent key escrow problem in ID-based cryptosystems: Since the TA generates private keys, the TA inherently knows the private keys of the users. Also, the private keys need to be sent to the user over a secure channel. Key revocation is another problem in ID-based cryptosystems, however, it can be solved by concatenating an expiration date to the identity of the user.

Digital signatures are the most commonly used public key cryptographic tool in online applications. They provide integrity, authentication, and non-repudiation. Integrity is ensuring that the message has not been changed by unauthorized entities; authentication is ensuring that the recipient can confirm the identity of the sender; and non-repudiation is ensuring that a communicating party cannot deny previous signatures and contracts.

In specific scenarios, some additional properties may be needed. For example, the *blindness* property is used in many applications such as electronic voting and electronic payment systems. By using blind signatures, the user is able to get a signature from an authority without revealing the actual message to the signer.

Signatures with *message recovery* can be preferred if bandwidth is a concern and the message length is small. In signature schemes with message recovery, the message is not transmitted together with the signature, and is recovered according to the verification process.

Signcription schemes, which combine the functionality of an encryption and a signature in a more efficient way, can be used if confidentiality is needed. Confidentiality is keeping the information secret from unauthorized entities.

The ElGamal signature scheme has been used as a key tool for constructing ID-based signature schemes. It has already been widely used for digital signatures and lots of variants were introduced after ElGamal's original proposal [?]. Horster et al. [?] integrated all these variants into a unified framework and obtained numerous variants of the original ElGamal signature scheme. ElGamal signatures with message recovery were proposed by Nyberg and Rueppel [?, ?]. Blind ElGamal like signatures were introduced by Camenisch [?]. Horster et al. [?, ?] also integrated these variants and produced numerous blind and message-recovery signatures.

1.1 Previous Work

In 1984, Shamir [?] introduced the concept of ID-based cryptography and he proposed an ID-based signature scheme based on the integer factorization problem. However, he could not find a practical ID-based encryption scheme. Later in 1986 Fiat and Shamir [?] proposed practical solutions for ID-based identification and signature schemes. In 1988, Guillou and Quisquater [?] proposed an ID-based signature scheme based on their zero knowledge protocol.

Menezes, Okamoto and Vanstone [?] used bilinear pairings to reduce the discrete logarithm problem on elliptic curves to the discrete logarithm problem in a finite field. This reduction was used to find weakness in elliptic curve cryptosystems and was called the MOV reduction. After Joux [?] discovery on how to utilize bilinear pairings in public cryptosystems, bilinear pairings were introduced as a cryptographic tool and ID-based cryptography became a very active area of research. In his paper, Joux proposed the first one-round protocol for tripartite Diffie-Hellman key exchange.

Finding a practical ID-based encryption scheme remained an open problem until Boneh and Franklin [?] proposed the first ID-based encryption scheme from bilinear pairings in Crypto 2001. In the same year Cocks [?] proposed an ID-based encryption scheme based on quadratic residues.

The first ID-based signature scheme from bilinear pairings was proposed by Sakai, Ohgisi and Kasahara [**?**] in 2000. However, they did not provide a security analysis. Paterson [**?**] proposed an ID-based signature scheme in 2002 which can be seen as an ID-based version of original the ElGamal signature. In Paterson's paper there was a brief security analysis but no rigorous proof. In 2002, Hess [**?**] proposed a provably secure ID-based signature scheme, which is secure against adaptively chosen message and fixed ID attacks. Yi [**?**] proposed a different ID-based signature scheme with brief security arguments in 2003.

In 2003, Cha and Cheon [**?**] proposed a provably secure ID-based signature scheme. They also provided a security definition for ID-based signature schemes and proved that their scheme is secure against adaptively chosen message and ID attacks.

In 2005, Barreto et al. [**?**] proposed a different provably secure ID-based signature scheme which is much more efficient than previously proposed schemes. They achieved this by changing the definitions of public and private keys. Their construction can be applied to most of the previous schemes to get more efficient signature schemes.

In 2006, Paterson and Schuldt [**?**] proposed an ID-based signature scheme. The novel feature of their signature is that the security proof of the signature does not depend on random oracles.

Sometimes a digital signature itself does not satisfy users' requirements and additional properties may be needed. If bandwidth is short and signature length is important, signature schemes with message recovery may be useful. RSA signatures [**?**] can be used with message recovery, since signature and encryption functions are inverse of each other. First signature scheme with message recovery based on the ElGamal signature was proposed by Nyberg and Rueppel [**?**] in 1993. Since then several other message recovery signatures have been proposed [**?**, **?**, **?**, **?**]

The first ID-based signature scheme giving message recovery was proposed by Zhang et al. [**?**] in 2005. They also gave a partial message recovery signature for

arbitrary length messages in their paper.

Tso et al. [?] proposed a different message recovery signature in 2007. They used the ideas of Barreto et al.[?] and changed the definitions of public and private keys. Thus, they obtained much more efficient signature schemes. They proved that their scheme is provably secure in the random oracle model.

Blind signatures are needed if a user wants to obtain a signature from an authority without revealing the actual message to the signer. Chaum [?] introduced the concept of blind signatures in 1982. Then in 1993, Camenisch et al. [?] showed that the ElGamal signature and its message recovery variant can be used to get a blind signature based on the discrete logarithm problem.

First ID-based blind signature was proposed by Zhang and Kim [?] in 2002. They improved the efficiency of their scheme and proposed a different blind signature scheme in 2003 [?]. Then in 2005, Huang et al. [?] proposed a more efficient blind signature.

In previous schemes two rounds were needed in the protocol between the user and the authority to get a blind signature. In 2007, Gao et al. [?] proposed a one-round ID-based blind signature scheme by sacrificing computational efficiency. Additionally they showed that their scheme is provably secure without using random oracles.

If confidentiality is also important, then signcryption schemes may be helpful. These schemes are much more efficient than the signature followed by encryption approach. Zheng [?] introduced the concept of signcryption in 1997. He coined the term signcryption to mean a primitive which simultaneously fulfills the job of a digital signature and encryption in a single step with a significantly lower cost. Since then several other signcryption schemes have been proposed [?, ?, ?, ?, ?, ?].

The first ID-based signcryption scheme was proposed by Malone-Lee [?] in 2002, however, there are security flows in the scheme. Libert and Quisquater [?] pointed out the problems in Malone-Lee's scheme and proposed a new ID-based signcryption scheme in 2003. Since then, several different signcryption schemes

are proposed by Boyen [**?**], Nalla and Reddy [**?**], McCullagh and Barreto [**?**], Chen and Malone-Lee [**?**]. In 2005, Barreto et al. [**?**] proposed the most efficient provably secure signcryption scheme so far by changing the definitions of the public and private keys.

1.2 Background

1.2.1 Bilinear Pairings

Cryptographers have been using bilinear pairings as a cryptographic tool since 1993. At first bilinear pairings were used to reduce the security of elliptic curve based cryptosystems. Menezes, Okamoto, Vanstone [**?**] and Frey, Rueck [] developed the MOV and FR attacks respectively. These attacks reduce the discrete logarithm problem (DLP) on certain elliptic curves to the DLP in a finite field.

Bilinear pairings' existence were believed to be a bad thing until Joux [**?**] discovered that they can be used as a cryptographic tool for designing cryptosystems. After that, ID-based cryptography became very popular in the literature based on bilinear pairings.

Let G_1 be a cyclic additive group of order q generated by P. Let G_2 be a cyclic multiplicative group of the same order. An admissible bilinear pairing is defined as $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

1. *Bilinearity:* $e(aR, bS) = e(R, S)^{ab}$ where $R, S \in G_1$ and $a, b \in \mathbb{Z}_q$. This can also be stated as $\forall R, S, T \in G_1$ $e(R + S, T) = e(R, T)e(S, T)$ and $e(R, S + T) = e(R, S)e(R, T)$

2. *Non-degeneracy:* The map e does not send all pairs in $G_1 \times G_1$ to the identity of G_2. That is $e(P, P) \neq 1$.

3. *Computability:* There exists an efficient algorithm to compute $e(R, S)$ for any $R, S \in G_1$

The Weil pairings and the Tate pairings of elliptic curves are examples of such admissible bilinear pairings.

1.2.2 Model of ID-based Signatures

An ID-based signature scheme consists of four algorithms:

- SETUP: The private key generator (PKG), a trusted authority, chooses the global secret key, computes the global public key and publishes it with other system parameters.

- EXTRACT: PKG verifies the user's identity and computes user's public and private key. The private key should be sent to the user over a secure channel after this phase.

- SIGN: An algorithm which takes the message m, user's private key, and other public parameters as input, and outputs the signature on m.

- VERIFY: An algorithm which takes the sender's identity, a signature, and other public system parameters as input, and outputs 1 if the signature is valid. Otherwise it outputs 0.

1.3 Problem Definition

The ElGamal signature scheme can be used as a key tool for constructing ID-based signature schemes. Most of the ID-based signatures in the literature [**?**, **?**, **?**, **?**] can be seen as variants of the basic ElGamal signature. However, providing a unified framework for these signatures has not investigated so far. The problem we address in this thesis is how to obtain a generalized ID-based signature scheme that provides a unified framework for previously proposed signatures.

We use the ElGamal signature to obtain an ID-based signature scheme. Then we show how the basic ID-based ElGamal signature scheme can be extended into a generalized ID-based signature. We discuss which variants are not possible and which variants are not secure in the ID-based setting. We also present some original variants which were not possible on the basic ElGamal scheme.

We extend our work to provide additional properties to our signature schemes. We investigate ID-based signatures providing message recovery. We show how the basic ElGamal signature with message recovery can be converted into an ID-based signature with message recovery. We extend our ID-based signature scheme into a generalized ID-based message recovery signature. We also present some original variants which were not possible in the non-ID-based setting. Then, we modify some of our signatures to get more efficient signature schemes.

ID-based blind signatures are also investigated in this thesis. We show how a modified blind ElGamal signature can be converted into an ID-based blind signature. We give a blindness proof for the resulting signature. We extend our basic ID-based blind signature scheme into a generalized ID-based blind signature. We discuss which variants are not possible in the ID-based setting. Then, we give an efficiency comparison of our signature with previously proposed blind signatures.

Lastly, ID-based blind signcryption schemes are investigated. We show how an ID-based signature scheme can be converted into an ID-based signcryption scheme. Then we generalize the idea and obtain signcryption schemes from our ID-based signatures.

1.4 Outline of the Thesis

In Chapter 2, we investigate the generalized ID-based ElGamal signatures. ElGamal signature and its variants are discussed in Section 2.1. We explain how to convert the original ElGamal signature into an ID-based signature scheme in

Section 2.2. We generalize the basic ID-based ElGamal signature scheme in Section 2.3. Security and efficiency of this generalized scheme are discussed section 2.4 and 2.5. We show how to embed previously proposed signatures into our generalized scheme in Section 2.6.

In Chapter 3, we investigate ID-based signature providing message recovery. ElGamal signatures with message recovery are discussed in Section 3.1. We describe the basic ID-based ElGamal signature with message recovery in Section 3.2. In Section 3.3, we describe the generalizations of the basic scheme. We modify some of these schemes and produce more efficient signatures in Section 3.4. We show how to embed previously proposed signatures into our generalized scheme in Section 3.5.

In Chapter 4, we discuss ID-based blind signatures. Blind ElGamal signatures and its generalizations are discussed in Section 4.1. We describe the basic ID-based blind signature scheme and its blindness proof in Section 4.2. In Section 4.3, we describe the generalizations of the basic scheme. We modify some of these schemes and produce more efficient signatures in Section 4.4. We give an efficiency comparison between some of our schemes and previously proposed signatures in Section 4.5.

In Chapter 5, we discuss ID-based signcryption schemes. We show how an ID-based signature scheme can be converted into an ID-based signcryption scheme and describe the basic ID-based signcryption scheme in Section 5.2. In Section 5.3, we describe the generalizations of the basic scheme. We modify some of these schemes and produce more efficient signcryption schemes in Section 5.4.

The thesis is concluded with a discussion of the proposed schemes in Chapter 6.

Chapter 2

Generalized ID-based ElGamal Signatures

In this chapter, we use the original ElGamal signature scheme to get an ID-based ElGamal signature scheme. Then we generalize the ID-based ElGamal signature scheme and propose various ID-based signature schemes based on the original one. We called our signature schemes as generalized ID-based ElGamal signatures. We use the ideas of Horster et al. [**?**] and bilinear pairings. Some of the variants found in this process were already proposed in literature but we also introduce some new types of variants that were not proposed before.

2.1 Background

2.1.1 ElGamal Signature Scheme

Let p be a large prime and g be a generator of $\mathbb{Z}_p^*$. The user chooses $\alpha \in \mathbb{Z}_{p-1}$ as his private key and then computes $\beta = g^\alpha \bmod p$ as his public key. The parameters p, g, and β are public whereas the user keeps α secret. To sign a message, the user generates a random $k \in_R \mathbb{Z}_{p-1}$. Then he computes $r = g^k \bmod p$ and $s = k^{-1}(m - r\alpha) \bmod (p-1)$. The (r, s) pair is the signature of message m. The

equation

$$m \equiv ks - r\alpha \pmod{p-1}$$

called signature equation and verification is done by checking the congruence $g^m \stackrel{?}{\equiv} \beta^r r^s \bmod p$. Security of ElGamal signature relies on the discrete logarithm problem (DLP) since solving α from β or s from r, m, β can be reduced to solving DLP in $\mathbb{Z}_p^*$.

2.1.2 The Meta-ElGamal Signature Scheme

Horster et al. [**?**] showed that many variations of the basic ElGamal signature are possible by modifying the signature equation. Instead of using ElGamal's original signature equation, one can use the general equation

$$A \equiv \alpha B + kC \bmod q$$

to obtain a signature, where (A, B, C) is a permutation of the parameters (m, r, s), q is a divisor of $p-1$, and g is an element in $\mathbb{Z}_p^*$ of order q. The signature can be verified by checking the equation:

$$g^A \stackrel{?}{\equiv} \beta^B r^C \pmod{p}$$

By these permutations six possible signatures can be obtained.

Different signature schemes can also be obtained by using different coefficients instead of just using the permutations of (m, r, s). The coefficients (A, B, C) can be chosen as a permutation of $(mr, s, 1)$, $(mr, ms, 1)$, $(mr, rs, 1)$, or $(mr, s, 1)$. Additionally the signs of (A, B, C) can be changed by multiplying them by ± 1. Then the signature equation will be

$$\pm A \equiv \pm \alpha B \pm kC \pmod{q}$$

where (A, B, C) is a permutation of the coefficients mentioned.

The generalization can be extended further by choosing A, B, C as general functions of m, r, s, instead of just products of two. The functions must be chosen carefully to guarantee the solvability and security. To guarantee solvability,

it is necessary that the parameter s can be extracted from the equation. To guarantee security, the parameters m, r, s have to occur in at least one of the three coefficients. Also, the insecure rs and ms variants should be avoided.

An insecure rs variant occurs if (A, B, C) is taken as a permutation of $(rs, m, 1)$: For some message m, an attacker chooses a random $c \in_R \mathbb{Z}_q^*$ and substitutes it for rs in the verification equation and computes r. Then he computes s as $s = cr^{-1}$. The (r, s) pair will be a valid signature for the message m.

An insecure ms variant occurs if (A, B, C) is a permutation of $(ms, r, 1)$: Assume that (r, s) is a valid signature observed by an adversary for some message m. For an arbitrary message m', the adversary computes s' as $s' = {m'}^{-1}ms$ and takes $r' = r$. Then (r', s') will be a valid signature for m'.

2.2 The Basic ID-based ElGamal Signature Scheme

An ID-based signature scheme consists of four algorithms: SETUP, EXTRACT, SIGN, and VERIFY. In SETUP, the trusted private key generator (PKG) chooses a secret as the global secret key and publishes the global public system parameters. In EXTRACT, the PKG verifies a user's identity and computes his private key. In SIGN, the user signs a message by using his private key. Finally in VERIFY, the verifier verifies the signature by using the public parameters and the signer's identity.

An ID-based signature scheme can be obtained from the original ElGamal signature scheme as follows:

- SETUP: Let G_1 be a cyclic additive group of order q generated by P. Let G_2 be a cyclic multiplicative group of the same order and $e : G_1 \times G_1 \to G_2$ be an admissible bilinear pairing. The PKG chooses $s \in_R \mathbb{Z}_q^*$ as the global

secret key and computes $P_{pub} = sP$ as the global public key. The PKG publishes system parameters $\langle G_1,\ G_2,\ e,\ P,\ P_{pub},\ H,\ H_1 \rangle$ where H and H_1 are secure hash functions.

- EXTRACT: PKG verifies the user's identity ID and computes $Q_{ID} = H_1(ID)$ and $S_{ID} = sQ_{ID}$ as user's public and private keys respectively.

- SIGN: To sign a message $m \in \mathbb{Z}_q$, a user with his private key S_{ID}, first chooses $k \in_R \mathbb{Z}_q$, then computes:

$$r = H(kP)$$
$$U = k^{-1}(mP - rS_{ID})$$

  The signature for the message m is (kP, U)

- VERIFY: Given ID, a message m, and a signature (kP, U), the signature is valid if the following equation holds.

$$e(U, kP)e(Q_{ID}, P_{pub})^r \stackrel{?}{=} e(P, P)^m \tag{2.1}$$

Correctness of the given scheme can be shown easily by using the bilinearity properties of e. Notice that if (kP, U) is a valid signature for m then we have:

$$\begin{aligned} e(U, kP)e(Q_{ID}, P_{pub})^r &= e(k^{-1}(mP - rS_{ID}), kP)e(Q_{ID}, P_{pub})^r \\ &= e(mP - rS_{ID}, P)e(rS_{ID}, P) \\ &= e(mP, P) \\ &= e(P, P)^m \end{aligned}$$

The above scheme is the ID-based version of the original ElGamal signature scheme. The conversion process, which will also be used for other signature equations, is described below:

In the original ElGamal scheme, the signature equation is $m \equiv \alpha r + ks \bmod (p-1)$ where $r = g^k$ and the signature is (r, s). Since additive elliptic curve

groups are used in ID-based structure, the signing equation and r will be slightly different. Signing equation for the ID-based ElGamal signature is:

$$mP = rS_{ID} + kU$$

Uppercase letters are used to denote elements of the elliptic curve group. S_{ID} is the private key of the user, so it is a natural replacement for α in the original scheme. U is a part of the signature and it is the replacement for s. We cannot use m directly since it is not a member of elliptic curve group; therefore mP is used to replace m. Here we can also use mQ_{ID} or mP_{pub} instead of mP and get a slightly different signature scheme.

A natural choice for r in the ID-based scheme is to compute r as $r = kP$ since r equals g^k in the original scheme. However, r must be an integer in $\mathbb{Z}_p$ in the signature equation, so we use a hash function and compute r as $r = H(kP)$. Additionally, since kP is needed for verification (**??**), the signature will be issued as (kP, U) instead of (r, U).

2.3 The Generalized ID-based ElGamal Signature and its Variants

We can generalize the above ID-based signature scheme by using the generalized signing equation

$$A = BS_{ID} + kC$$

where (A, B, C) is a permutation of the parameters (m, r, U), instead of the basic equation $mP = rS_{ID} + kU$. Note that, not all the permutations generate useful variants. We should consider that U is a member of elliptic curve group, and $m, r \in \mathbb{Z}_p$. Accordingly, A and C should be members of the elliptic curve group, but not B. Also note that, we can use mP and rP instead of m and r , in cases where they need to be members of the elliptic curve group.

We get four variants by simply permuting the elements of (m, r, U). The

signing equation for these variants are:

$$mP = rS_{ID} + kU \tag{2.2}$$

$$U = rS_{ID} + kmP \tag{2.3}$$

$$U = mS_{ID} + krP \tag{2.4}$$

$$rP = mS_{ID} + kU \tag{2.5}$$

Note that, the two variants where U is a coefficient of S_{ID} do not produce useful signing equations.

In the variants where kP is not needed for verification, r can be computed as $e(P,P)^k$ and the signature for m will be (r,U). This has the advantage that we can get rid of one pairing operation in the verification phase. Additionally, since the signer knows k, he can compute $e(P,P)^k$ without any pairing computation. As can be seen in Table **??**, r is taken as $e(P,P)^k$ in (**??**) and (**??**). Note that, in (**??**) and (**??**), we need the value of kP for verification. In that case r will be computed as $r = H(kP)$ and the signature for m will be (kP,U). We can also compute r as $H(m,kP)$ instead of $H(kP)$ or $e(P,P)^k$. In that case, m does not need to occur in the signing equations.

We can generate more variants by using different permutations. Instead of choosing (A,B,C) as a permutation of (m,r,U), we can also choose them as a permutation of $(mr,U,1)$, $(mr,mU,1)$ and $(mr,rU,1)$. Signs of A,B, and C can be changed by multiplying them by ± 1. We can also use a general function $f(m,r)$ instead of just product mr. Note that, unlike the original ElGamal variants, we cannot choose (A,B,C) as a permutation of $(mU,rU,1)$, since we cannot extract U from the signing equation in these variants. The signature equations for these ID-based ElGamal variants can be found in Table **??**.

The verification equations and other details for all signatures are summarized in Table **??**. Group I lists the variants that are obtained by permuting (m,r,U) and $(1,r,U)$. Group II lists the variants that are obtained by permuting $(mr,U,1)$. Group III lists the variants that are obtained by permuting $(mr,mU,1)$. Group IV lists the variants that are obtained by permuting $(mr,rU,1)$ and $(r,rU,1)$. Group V shows the rU variants discussed in Section **??**.

No.	A	B	C	ElGamal Variant	ID-Based Signature
ID I.1	m	r	U	$m \equiv \alpha r + ks$	$mP = rS_{ID} + kU$
ID I.2	r	m	U	$r \equiv \alpha m + ks$	$rP = mS_{ID} + kU$
ID I.3	U	r	m	$s \equiv \alpha r + km$	$U = rS_{ID} + kmP$
ID I.4	U	m	r	$s \equiv \alpha m + kr$	$U = mS_{ID} + rkP$
ID II.1	1	mr	U	$1 \equiv mr\alpha + ks$	$P = mrS_{ID} + kU$
ID II.2	mr	1	U	$mr \equiv \alpha + ks$	$mrP = S_{ID} + kU$
ID II.3	U	mr	1	$s \equiv mr\alpha + k$	$U = mrS_{ID} + kP$
ID II.4	U	1	mr	$s \equiv \alpha + kmr$	$U = -S_{ID} - mrkP$
ID III.1	1	mr	mU	$1 \equiv mr\alpha + kms$	$P = mrS_{ID} + mkU$
ID III.2	mr	1	mU	$mr \equiv \alpha + kms$	$mrP = S_{ID} + kmU$
ID III.3	mU	mr	1	$ms \equiv mr\alpha + k$	$mU = mrS_{ID} + kP$
ID III.4	mU	1	mr	$ms \equiv \alpha + kmr$	$mU = S_{ID} + mrkP$
ID IV.1	mr	1	Ur	$mr \equiv \alpha + krs$	$mrP = S_{ID} + rkU$
ID IV.2	1	mr	Ur	$1 \equiv mr\alpha + krs$	$P = mrS_{ID} + rkU$
ID IV.3	Ur	1	mr	$rs \equiv \alpha + mrk$	$rU = S_{ID} + mrkP$
ID IV.4	Ur	mr	1	$rs \equiv mr\alpha + k$	$rU = mrS_{ID} + kP$

Table 2.1: ElGamal variants and the corresponding ID-based ElGamal signature equations.

Finally group VI shows the variants discussed in Section **??** that were not possible on the basic ElGamal signatures.

2.4 Security Analysis of Proposed Schemes

The generalized ElGamal signature schemes of Horster et al. [**?**] are believed to be secure except two insecure variants. The two insecure variants in the generalized ElGamal signature schemes are the rs and ms variants as discussed in Section **??**. The corresponding ID-based variants are the rU and mU variants. These variants occur if (A, B, C) is a permutation of $(rU, m, 1)$ or $(mU, r, 1)$, respectively.

The mU variants are completely insecure and the attack works similar to the attack for the ms variant of the basic ElGamal signature: Assume that the (r, U) pair is a valid signature observed by the adversary for message m. For an arbitrary message m', the adversary computes $U' = {m'}^{-1}mU$ and uses $r' = r$.

	Signature equation	Verification equation
V.1	$mP = S_{ID} + rkU$	$e(U, kP)^r e(Q_{ID}, P_{pub}) = e(P, P)^m$
V.2	$P = mS_{ID} + rkU$	$e(U, kP)^r e(Q_{ID}, P_{pub})^m = e(P, P)$
V.3	$rU = -mS_{ID} + kP$	$e(U, rP)e(Q_{ID}, P_{pub})^m = r$
V.4	$rU = -S_{ID} + mkP$	$e(U, rP)e(Q_{ID}, P_{pub}) = r^m$
V.5	$P = S_{ID} + rkU$	$e(U, kP)^r e(Q_{ID}, P_{pub}) = e(P, P)$
V.6	$rU = -S_{ID} + kP$	$e(U, rP)e(Q_{ID}, P_{pub}) = r$

Table 2.2: the rU variants

Then (r', U') pair will be a valid signature for m'.

This is not always the case for the rU variants; the attack on the basic ElGamal rs variants does not work for two of the four ID-based rU variants. Signature and verification equation for the rU variants can be seen in Table **??**.

In Table **??**, the variants V.3, V.4 and V.6 are insecure. The attack for these rU variants works as follows: For an arbitrary message m, the adversary chooses $C \in_R G_1$. Then he substitutes $e(C, P)$ for $e(U, rP)$ in the verification equation and computes r. After that, he computes $U = r^{-1}C$. The (r, U) pair will be a valid signature for the message m.

The variants V.1, V.2 and V.5 in Table **??** seem to be secure since an attacker cannot extract r from the verification equation. Therefore, we have three more ID-based signatures from the rU variants.

2.5 Efficiency of the Proposed Schemes

As the main computational cost, we consider the number of bilinear pairings, modular exponentiations, and scalar multiplications in elliptic curve group. We assume the value of $e(P, P)$ is precomputed by every party.

Computing a signature requires a scalar multiplication in G_1 or an exponentiation in G_2 depending on how r is computed, as well as on or two scalar

multiplication in G_1 depending on the signature equation.

The cost of verifying a signature will be dominated by the pairing computations, which is the most expensive operation. Two or three pairing computations are needed to verify a signature depending on the signing equation. Note that, the value $e(Q_{ID}, P_{pub})$ is fixed for a particular user, so it needs to be computed once for each user.

More efficient variants can be obtained by modifying the generalized signature equation (**??**) as

$$A = BS_{ID} + kCS_{ID} \tag{2.6}$$

Note that, this kind of generalization is not possible for the basic ElGamal signature because when k and α are used together we cannot extract s from the signing equation.

By the help of bilinear pairings we can solve U from the signature equation (**??**) if we choose (A, B, C) as a permutation of (m, r, U), $(mr, U, 1)$ or $(m, rU, 1)$. Note that B and C cannot be a member of the elliptic curve group; hence U should be in A's position. So we get six more variants by using equation (**??**). These variants are:

$$\begin{aligned} U &= rS_{ID} + kmS_{ID} \\ U &= mS_{ID} + krS_{ID} \\ U &= rmS_{ID} + kS_{ID} \\ U &= S_{ID} + kmrS_{ID} \\ rU &= mS_{ID} + kS_{ID} \\ rU &= S_{ID} + kmS_{ID} \end{aligned}$$

The value of kQ_{ID} will be needed for verification. Therefore r is computed as $r = H(kQ_{ID})$ for these variants. For a message m the signature will be (kQ_{ID}, U). We can also compute r as $r = H(m, kQ_{ID})$ and remove m from the signing equations. Group VI of Table **??** shows the verification equations and other details for these schemes.

As observed by Barreto et al. [**?**], the number of pairing operations needed

can be reduced further by changing the definitions of S_{ID} and Q_{ID} as

$$Q_{ID} = (H_1(ID) + s)P,$$
$$S_{ID} = (H_1(ID) + s)^{-1}P.$$

For instance, for the signature (r, U), $r = e(P, P)^k$, $U = (k + mr)S_{ID}$, the verification equation becomes

$$r = e(U, Q_{ID})e(P, P)^{-mr},$$

and the number of pairing evaluations needed is reduced to one.

A similar modification can also be applied to the other signature schemes discussed in this thesis to reduce the number of pairing evaluations in each verification.

2.6 Embedding Previously Known ID-based Signatures

Recently many ID-based signature schemes have been proposed. Most of these signatures [**?**, **?**, **?**, **?**] can be seen as special instances of our generalized scheme:

- In Paterson's scheme [**?**], the signature (kP, U) is computed as

  $$r = H(kP)$$
  $$U = k^{-1}(H_2(m)P + rS_{ID})$$

  where H_2 is a secure hash function. Paterson's scheme is equivalent to ID I.1 of Table **??** where a second hash function H_2 is used for message digest.

- In Cha-Cheon's scheme [**?**], the signature (kQ_{ID}, U) is computed as

  $$r = H(m, kQ_{ID})$$
  $$U = (r + k)S_{ID}$$

  Cha-Cheon's scheme is the same as ID VI.7.

No.	r	U	Signature	Verification
ID I.1	$r = H(kP)$	$U = k^{-1}(mP - rS_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^r = e(P, P)^m$
ID I.2	$r = H(kP)$	$U = k^{-1}(rP - mS_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^m = e(P, P)^r$
ID I.3	$r = e(P, P)^k$	$U = kmP - rS_{ID}$	(r, U)	$e(U, P)e(Q_{ID}, P_{pub})^r = r^m$
ID I.4	$r = e(P, P)^k$	$U = rkP - mS_{ID}$	(r, U)	$e(U, P)e(Q_{ID}, P_{pub})^m = r^r$
ID I.5	$r = H(m, kP)$	$U = k^{-1}(P - rS_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^r = e(P, P)$
ID I.6	$r = H(m, kP)$	$U = k^{-1}(rP - S_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub}) = e(P, P)^r$
ID I.7	$r = H(m, kP)$	$U = kP - rS_{ID}$	(kP, U)	$e(U, P)e(Q_{ID}, P_{pub})^r = e(P, kP)$
ID I.8	$r = H(m, kP)$	$U = rkP - S_{ID}$	(kP, U)	$e(U, P)e(Q_{ID}, P_{pub}) = e(P, kP)^r$
ID II.1	$r = H(kP)$	$U = k^{-1}(P - mrS_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^{mr} = e(P, P)$
ID II.2	$r = H(kP)$	$U = k^{-1}(-S_{ID} + mrP)$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub}) = e(P, P)^{mr}$
ID II.3	$r = e(P, P)^k$	$U = kP - mrS_{ID}$	(r, U)	$e(U, P)e(Q_{ID}, P_{pub})^{mr} = r$
ID II.4	$r = e(P, P)^k$	$U = mrkP - S_{ID}$	(r, U)	$e(U, P)e(Q_{ID}, P_{pub}) = r^{mr}$
ID III.1	$r = H(kP)$	$U = k^{-1}(m^{-1}P - rS_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^r = e(P, P)^{m^{-1}}$
ID III.2	$r = H(kP)$	$U = k^{-1}(rP - m^{-1}S_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^{m^{-1}} = e(P, P)^r$
ID III.3	$r = e(P, P)^k$	$U = m^{-1}kP - rS_{ID}$	(r, U)	$e(U, P)e(Q_{ID}, P_{pub})^r = r^{m^{-1}}$
ID III.4	$r = e(P, P)^k$	$U = rkP - m^{-1}S_{ID}$	(r, U)	$e(U, P)e(Q_{ID}, P_{pub})^{m^{-1}} = r^r$
ID IV.1	$r = H(kP)$	$U = k^{-1}(mP - r^{-1}S_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^{r^{-1}} = e(P, P)^m$
ID IV.2	$r = H(kP)$	$U = k^{-1}(r^{-1}P - mS_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^m = e(P, P)^{r^{-1}}$
ID IV.3	$r = e(P, P)^k$	$U = mkP - r^{-1}S_{ID}$	(r, U)	$e(U, P)e(Q_{ID}, P_{pub})^{r^{-1}} = r^m$
ID IV.4	$r = e(P, P)^k$	$U = r^{-1}kP - mS_{ID}$	(r, U)	$e(U, P)e(Q_{ID}, P_{pub})^m = r^{r^{-1}}$
ID IV.5	$r = H(m, kP)$	$U = k^{-1}(P - r^{-1}S_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub})^{r^{-1}} = e(P, P)$
ID IV.6	$r = H(m, kP)$	$U = k^{-1}(r^{-1}P - S_{ID})$	(kP, U)	$e(U, kP)e(Q_{ID}, P_{pub}) = e(P, P)^{r^{-1}}$
ID IV.7	$r = H(m, kP)$	$U = kP - r^{-1}S_{ID}$	(kP, U)	$e(U, P)e(Q_{ID}, P_{pub})^{r^{-1}} = e(P, kP)$
ID IV.8	$r = H(m, kP)$	$U = r^{-1}kP - S_{ID}$	(kP, U)	$e(U, P)e(Q_{ID}, P_{pub}) = e(P, kP)^{r^{-1}}$
ID V.1	$r = H(kP)$	$U = k^{-1}r^{-1}(mP - S_{ID})$	(kP, U)	$e(U, kP)^r e(Q_{ID}, P_{pub}) = e(P, P)^m$
ID V.2	$r = H(kP)$	$U = k^{-1}r^{-1}(P - mS_{ID})$	(kP, U)	$e(U, kP)^r e(Q_{ID}, P_{pub})^m = e(P, P)$
ID V.3	$r = H(m, kP)$	$U = k^{-1}r^{-1}(P - S_{ID})$	(kP, U)	$e(U, kP)^r e(Q_{ID}, P_{pub}) = e(P, P)$
ID VI.1	$r = H(kQ_{ID})$	$U = (r + km)S_{ID}$	(kQ_{ID}, U)	$e(U, P) = e((r + km)Q_{ID}, P_{pub})$
ID VI.2	$r = H(kQ_{ID})$	$U = (m + kr)S_{ID}$	(kQ_{ID}, U)	$e(U, P) = e((kr + m)Q_{ID}, P_{pub})$
ID VI.3	$r = H(kQ_{ID})$	$U = (rm + k)S_{ID}$	(kQ_{ID}, U)	$e(U, P) = e((rm + k)Q_{ID}, P_{pub})$
ID VI.4	$r = H(kQ_{ID})$	$U = (1 + kmr)S_{ID}$	(kQ_{ID}, U)	$e(U, P) = e((1 + mkr)Q_{ID}, P_{pub})$
ID VI.5	$r = H(kQ_{ID})$	$U = r^{-1}(m + k)S_{ID}$	(kQ_{ID}, U)	$e(U, P)^r = e((m + k)Q_{ID}, P_{pub})$
ID VI.6	$r = H(kQ_{ID})$	$U = r^{-1}(1 + kmS_{ID})$	(kQ_{ID}, U)	$e(U, P)^r = e((mk + 1)Q_{ID}, P_{pub})$
ID VI.7	$r = H(m, kQ_{ID})$	$U = (r + k)S_{ID}$	(kQ_{ID}, U)	$e(U, P) = e((r + k)Q_{ID}, P_{pub})$
ID VI.8	$r = H(m, kQ_{ID})$	$U = r^{-1}(1 + k)S_{ID}$	(kQ_{ID}, U)	$e(U, P)^r = e((1 + k)Q_{ID}, P_{pub})$

Table 2.3: The generalized ID-based ElGamal signatures and their verification equations.

- In Yi's scheme [**?**], the signature (kP, U) is computed as

$$r = H(m, kP)$$
$$U = kP_{pub} + rS_{ID}$$

  Yi's scheme is equivalent to ID I.7, where, P_{pub} is used instead of P and the verification procedure is modified accordingly.

- In Hess's scheme [**?**], the signature (v, U) is computed as

$$r = e(P_1, P)^k$$
$$v = H(m, r)$$
$$U = kP_1 + vS_{ID}$$

  where P_1 is an arbitrary point on the curve. Hess's scheme can be converted into ID II.3 with $P_1 = P$ and using mr instead of $v = H(m, r)$. Besides, in Hess's scheme, verification takes an extra step for checking $v \stackrel{?}{\equiv} H(m, r)$.

Chapter 3

Generalized ID-Based ElGamal Signatures with Message Recovery

In this chapter, we introduce the concept of generalized ID-based ElGamal signatures with message recovery and show that the previously proposed signature schemes are special instances of this generalized scheme. The generalized scheme also yields many new ID-based signatures with message recovery that have not been explored before.

3.1 Background

3.1.1 ElGamal Signature Scheme with Message Recovery

Nyberg and Rueppel showed that the ElGamal signatures can be extended to provide message recovery. The extension is done as follows: Let p be a large prime, q a divisor of $p-1$, and g an element in $\mathbb{Z}_p^*$ of order q. The user chooses $\alpha \in \mathbb{Z}_q$ as his private key and $\beta = g^\alpha \bmod p$ as his public key. To sign a message

$m \in \mathbb{Z}_p$, the user first generates a random number $k \in_R \mathbb{Z}_q^*$. Then he computes:

$$r = mg^{-k} \bmod p$$
$$s = k^{-1}(1 + r\alpha) \bmod q$$

The (r, s) pair is the signature of message m. The equation,

$$1 = r\alpha + ks \bmod q \tag{3.1}$$

is called the signature equation and the message m can be recovered by computing $m = g^{s^{-1}}\beta^{rs^{-1}}r \bmod p$. We call this scheme as the basic ElGamal message recovery scheme.

Note that, in the above scheme computation of the signature and message recovery involve inversion of the elements in $\mathbb{Z}_q$. Nyberg and Rueppel showed that it is also possible to get a signature without inversions. Signature computation and verification can be done without inverses by changing the signature equation as:

$$s = -\alpha r + k \bmod q.$$

The message m can now be recovered as $m = g^s\beta^r r \bmod p$ without any inversions.

3.1.2 Generalized ElGamal Signatures with Message Recovery

Horster et al. [**?**] showed that many variations of the basic ElGamal message recovery scheme are possible by modifying the signature equation (**??**). One can use the general equation

$$A = \alpha B + kC \bmod q$$

to obtain a signature, where (A, B, C) is a permutation of the parameters $(1, r, s)$. The parameter r can be computed as $r = g^{-k}m$ or $r = d(m, g^k)$ with a suitable function $d : \mathbb{Z}_p^2 \to \mathbb{Z}_p$ where $d^{-1}(r, g^k) = m$. The message m can be recovered from the signature (r, s) by computing

$$m = d^{-1}(r, g^{AC^{-1}\beta^{-BC^{-1}}} \bmod p).$$

The consistency of m should be verified by checking if m satisfies a certain redundancy scheme as explained in Section **??**.

Different signature schemes can be obtained by using different coefficients instead of just using the permutations of $(1, r, s)$. The coefficients (A, B, C) can also be chosen as a permutation of $(1, r, rs)$ or $(1, s, rs)$. Additionally, the signs of (A, B, C) can be changed by multiplying them by ± 1.

The generalization can be extended further by choosing A, B, C as general functions of r, s. In that case one of the functions should be chosen as 1 to get efficient variants. Additionally, suitable functions should be chosen to guarantee solvability of the parameter s. To guarantee security, the parameters r, s have to occur in at least one of the three coefficients. Also, the insecure rs variant should be avoided.

An insecure rs variant occurs if (A, B, C) is taken as a permutation of $(1, 1, rs)$: For some message m, an attacker chooses a random $c \in_R \mathbb{Z}_q^*$ and substitutes it for rs and computes g^{-k} from the verification equation. Then he computes first r from g^{-k} and then computes s as $s = cr^{-1}$. The (r, s) pair will be a valid signature for the message m.

3.2 Basic ID-based ElGamal Signatures with Message Recovery

An ID-based signature scheme consists of four algorithms: Setup, Extract, Sign, and Verify. In Setup, the PKG, chooses a secret as the global secret key and publishes the global public system parameters. In Extract, the PKG verifies a user's identity and computes his private key. In Sign, the user signs a message by using his private key. Finally in Verify, the verifier verifies the signature and recovers the message by using the public parameters and the signer's identity.

An ID-based message recovery signature scheme can be obtained from the

original ElGamal signature scheme as follows:

- SETUP: Let G_1 be cyclic additive group of order q generated by P. Let G_2 be a cyclic multiplicative group of the same order and $e : G_1 \times G_1 \rightarrow G_2$ be an admissible bilinear pairing. The PKG chooses $s \in_R \mathbb{Z}_q^*$ as the global secret key and computes $P_{pub} = sP$ as the global public key. The PKG publishes system parameters $\langle G_1,\ G_2,\ e,\ P,\ P_{pub},\ H_1 \rangle$ where H_1 is a secure hash function.

- EXTRACT: PKG verifies the user's identity ID and computes $Q_{ID} = H_1(ID)$ and $S_{ID} = sQ_{ID}$ as user's public and private keys respectively.

- SIGN: To sign a message $m \in \mathbb{Z}_q$, a user with his private key S_{ID}, first chooses $k \in_R \mathbb{Z}_q$, then computes:

$$r = e(P, P)^k \oplus m$$
$$U = k(P - rS_{ID})$$

  The signature for the message m is (kP_{pub}, r, U)

- VERIFY: Given ID, and a signature (kP_{pub}, r, U), the message can be recovered as:

$$m = r \oplus (e(U, P)e(Q_{ID}, kP_{pub})^r)$$

Correctness of the given scheme can be shown easily by using the bilinearity properties of e. Notice that if (kP_{pub}, r, U) is a valid signature for m then we have:

$$\begin{aligned} e(U, P)e(Q_{ID}, kP_{pub})^r &= e(k(P - rS_{ID}), P)e(Q_{ID}, kP_{pub})^r \\ &= e(kP - krS_{ID}, P)e(krS_{ID}, P) \\ &= e(kP, P) \\ &= e(P, P)^k \end{aligned}$$

3.2.1 Consistency Checking for the Message

In order to prevent a random (r, s) pair being accepted as a valid signature, consistency of the message should be checked with a given redundancy scheme. Abe and Okamoto [**?**] proposed such a redundancy encoding for their message recovery signature which can also be used in our scheme: Let $|q|$ denote the length of q in bits. Let $[m']^{k_1}$ denote the most significant k_1 bits of m' and $[m']_{k_2}$ denote the least significant k_2 bits of m'. Instead of computing r as $r = e(P, P)^k \oplus m$, first compute

$$m' = F_1(m) \| (F_2(F_1(m)) \oplus m),$$

where $F_1 : \{0,1\}^{k_2} \rightarrow \{0,1\}^{k_1}$ and $F_2 : \{0,1\}^{k_1} \rightarrow \{0,1\}^{k_2}$ are secure hash functions; and compute

$$r = e(P, P)^k \oplus m'.$$

Then the message m with length $|k_2|$ can be recovered as

$$m = [m']_{k_2} \oplus F_2([m']^{k_1}).$$

Consistency of m can be verified by checking $[m']^{k_1} \stackrel{?}{=} F_1(m)$. The advantage of using Abe and Okamoto's redundancy encoding is that F_1 and F_2 can be seen as random oracles so m' will be a random value independent from m.

3.3 The Generalized ID-based Message Recovery Signatures

We can generalize the above signature scheme by using the generalized signing equation

$$A = S_{ID}B + kC \tag{3.2}$$

where (A, B, C) is a permutation of the parameters $(1, r, U)$. Note that, the variants where U is a coefficient of S_{ID} do not produce useful signing equations. Also note that, P and rP are used instead of 1 and r in cases where they need to be members of the elliptic curve group.

No.	r	U	Signature	Message Recovery
MR I.1	$r = e(P,P)^k \oplus m$	$U = kP - krS_{ID}$	(kP_{pub}, r, U)	$m = r \oplus (e(U,P)e(Q_{ID}, kP_{pub})^r)$
MR I.2	$r = e(P,P)^k \oplus m$	$U = kP - rS_{ID}$	(r, U)	$m = r \oplus (e(U,P)e(Q_{ID}, P_{pub})^r)$
MR I.3	$r = e(P,P)^k \oplus m$	$U = krP - S_{ID}$	(r, U)	$m = r \oplus (e(U,P)^{r^{-1}} e(Q_{ID}, P_{pub})^{r^{-1}})$
MR I.4	$r = e(P,P)^k \oplus m$	$U = krP - kS_{ID}$	(kP_{pub}, r, U)	$m = r \oplus (e(U,P)^{r^{-1}} e(Q_{ID}, kP_{pub})^{r^{-1}})$
MR II.1	$r = e(P,P)^k \oplus m$	$U = r^{-1}kP - kS_{ID}$	(kP_{pub}, r, U)	$m = r \oplus (e(U,P)^r e(Q_{ID}, kP_{pub})^r)$
MR II.2	$r = e(P,P)^k \oplus m$	$U = r^{-1}kP - S_{ID}$	(r, U)	$m = r \oplus (e(U,P)^r e(Q_{ID}, P_{pub})^r)$
MR II.3	$r = e(P,P)^k \oplus m$	$U = kP - r^{-1}S_{ID}$	(r, U)	$m = r \oplus (e(U,P)e(Q_{ID}, P_{pub})^{r^{-1}})$
MR II.4	$r = e(P,P)^k \oplus m$	$U = kP - r^{-1}kS_{ID}$	(kP_{pub}, r, U)	$m = r \oplus (e(U,P)e(Q_{ID}, kP_{pub})^{r^{-1}})$
MR III.1	$r = e(P,P)^k \oplus m$	$U = r^{-1}k(P - S_{ID})$	(kP_{pub}, r, U)	$m = r \oplus (e(U,P)^r e(Q_{ID}, kP_{pub}))$

Table 3.1: The generalized ID-based ElGamal signatures with message recovery.

We get four variants by permuting the elements of $(1, r, U)$. The signing equation for these variants are:

$$P = rS_{ID} + k^{-1}U \tag{3.3}$$

$$U = rS_{ID} + kP \tag{3.4}$$

$$U = S_{ID} + krP \tag{3.5}$$

$$rP = S_{ID} + k^{-1}U \tag{3.6}$$

In (**??**) and (**??**) the signature for m will be (r, U) and we can recover m without any extra information. However, in (**??**) and (**??**) we need the value of kP_{pub} for verification, and the signature will be (kP_{pub}, r, U).

More variants can be generated by using different permutations. Instead of choosing (A, B, C) as a permutation of $(1, r, U)$, we can also choose them as a permutation of $(1, r, rU)$. Also, signs of A, B and C can be changed by multiplying them by ± 1. Note that, unlike the generalized ElGamal message recovery signatures, we cannot choose (A, B, C) as a permutation of $(1, U, rU)$, since we cannot extract U from the signing equation.

The verification equations and other details for these signatures are summarized in Table **??**. Group I lists the variants that are obtained by permuting $(1, r, U)$ and Group II lists the variants obtained by permuting $(1, r, rU)$. Group III is the secure $(1, 1, rU)$ variant which is discussed in Section **??**.

3.3.1 Generalized Partial Message Recovery Signatures

In the above signature schemes, length of the message is fixed. If Abe and Okamoto's redundancy encoding is used, then $|m| = k_2$. Here we show how one of the previous schemes can be modified to allow arbitrary length messages by splitting the message m into two parts called m_1 and m_2. The first part m_1 is of arbitrary length and is given with the signature (r, U). The second part m_2 has a fixed length and is recovered from the signature.

As an example, consider MR I.2 of Table **??**. To sign a message $m = m_1 \| m_2$ with $m_2 \in \mathbb{Z}_q$, a user with his private key S_{ID}, first chooses $k \in_R \mathbb{Z}_q$, then computes:

$$r = e(P, P)^k \oplus m_2$$
$$U = kP - m_1 r S_{ID}$$

The signature for the message m is (m_1, r, U). Note that, a general function $f(m_1, r)$ can be used instead of the product $m_1 r$.

To verify a given signature (m_1, r, U), the message can be recovered as:

$$m_2 = r \oplus (e(U, P) e(Q_{ID}, P_{pub})^{m_1 r})$$
$$m = m_1 \| m_2$$

Correctness of this scheme can easily be shown by using the bilinearity properties of e. Consistency of m should be verified by checking if m satisfies a certain redundancy scheme.

3.3.2 Security of the Signatures

Similar to the meta-ElGamal signature schemes with message recovery [**?**], generalized ID-based signatures with message recovery are generally secure except the insecure rU variants. These variants occur if (A, B, C) is either $(rU, 1, 1)$ or

$(1, 1, rU)$. Signing equations for these variants are:

$$rU = -S_{ID} + kP \tag{3.7}$$

$$P = S_{ID} + rk^{-1}U \tag{3.8}$$

In (**??**) the message m should satisfy the verification equation

$$m = r \oplus (e(U, P)^r e(Q_{ID}, P_{pub}))$$

This signature is not secure and the rU attack for this signature works as follows: For arbitrary message m, the adversary chooses $T \in_R G_1$. The random T will be used instead of rU so the adversary substitutes $e(T, P)$ for $e(U, P)^r$ and computes $e(P, P)^k$ as

$$e(P, P)^k = e(T, P)e(Q_{ID}, P_{pub})$$

Then he computes r as $r = e(P, P)^k \oplus m$. After that, he computes $U = r^{-1}C$. The (r, U) pair will be a valid signature for the message m.

The verification equation for the signature obtained from (**??**) is

$$m = r \oplus (e(U, P)^r e(Q_{ID}, kP_{pub}))$$

This signature seems to be secure and the rU attack does not work because the verification equation contains kP_{pub}. Therefore, an attacker cannot extract r from the verification equation.

3.4 More Efficient Signatures

As the main computational cost, we consider the number of bilinear pairings, modular exponentiations, and scalar multiplications in elliptic curve group. We assume the value of $e(P, P)$ is precomputed by every party.

Computing a signature requires one or two scalar multiplications in G_1 depending on how the signature equation is defined, as well as an exponentiation in G_2. Pairing evaluation is not needed to sign a message.

The cost of verifying a signature will be dominated by the pairing computations, which is the most expensive operation. Two pairing computations are needed to verify a signature. Note that, in some of the proposed schemes (MR I.2, MR I.3, MR II.2, MR II.3), the value $e(Q_{ID}, P_{pub})$ is used, which is fixed for a particular user and needs to be computed only once for each user.

The number of pairing operations can be reduced to one by changing the definitions of S_{ID} and Q_{ID} as in [**?**]. If we define

$$Q_{ID} = (H_1(ID) + s)P$$
$$S_{ID} = (H_1(ID) + s)^{-1}P,$$

the number of pairing evaluations can be reduced to one. Note that, Q_{ID} can be computed by anyone, since the value of sP is public, but S_{ID} cannot be computed without knowing the value of s.

We can get efficient variants by changing the definitions of S_{ID} and Q_{ID} in four of the proposed schemes. These schemes are MR I.2, MR I.3, MR II.2, MR II.3 of Table **??**. The computation of r should also be changed to increase the efficiency. Instead of computing r as $r = e(P, P)^k \oplus m$, r will be computed as

$$r = e(P, Q_{ID})^k \oplus m$$

This modification does not affect the efficiency of signature computation, since the value $e(P, Q_{ID})$ can be precomputed by the sender.

As an example, consider the modified version of MR I.2 where $U = kP - rS_{ID}$. The message m can be recovered from the signature (r, U) as,

$$m = r \oplus (e(U, Q_{ID})e(P, P)^r).$$

The verification equations and other details of the efficient versions of MR I.2, MR I.3, MR II.2, MR II.3 modified in this fashion are given in Group IV of Table **??**.

Further variants with a reduced signing cost can be obtained by modifying the generalized signature equation as,

$$A = BS_{ID} + kCS_{ID}. \tag{3.9}$$

No.	r	U	Signature	Message Recovery
MR IV.1	$r = e(P, Q_{ID})^k \oplus m$	$U = kP - rS_{ID}$	(r, U)	$m = r \oplus (e(U, Q_{ID})e(P, P)^r)$
MR IV.2	$r = e(P, Q_{ID})^k \oplus m$	$U = krP - S_{ID}$	(r, U)	$m = r \oplus (e(U, Q_{ID})^{r^{-1}} e(P, P)^{r^{-1}})$
MR IV.3	$r = e(P, Q_{ID})^k \oplus m$	$U = r^{-1}kP - S_{ID}$	(r, U)	$m = r \oplus (e(U, Q_{ID})^r e(P, P)^r)$
MR IV.4	$r = e(P, Q_{ID})^k \oplus m$	$U = kP - r^{-1}S_{ID}$	(r, U)	$m = r \oplus (e(U, Q_{ID})e(P, P)^{r^{-1}})$
MR V.1	$r = e(P, P)^k \oplus m$	$U = (k + r)S_{ID}$	(r, U)	$m = r \oplus (e(U, Q_{ID})e(P, P)^{-r})$
MR V.2	$r = e(P, P)^k \oplus m$	$U = (1 + kr)S_{ID}$	(r, U)	$m = r \oplus (e(U, Q_{ID})^{r^{-1}} e(P, P)^{-r^{-1}})$
MR V.3	$r = e(P, P)^k \oplus m$	$U = r^{-1}(k + r)S_{ID}$	(r, U)	$m = r \oplus (e(U, Q_{ID})^r e(P, P)^{-r})$
MR V.4	$r = e(P, P)^k \oplus m$	$U = r^{-1}(1 + kr)S_{ID}$	(r, U)	$m = r \oplus (e(U, Q_{ID})e(P, P)^{-r^{-1}})$

Table 3.2: Efficient ID-based signatures with message recovery.

Note that, this kind of generalization is not possible over the basic ElGamal signatures, because when k and α are used together, we cannot extract s from the signing equation.

By the help of bilinear pairings we can extract U from the signing equation (**??**), if U is in A's position. We can get four more efficient variants whose signing equations are:

$$U = (k + r)S_{ID}$$
$$U = (1 + kr)S_{ID}$$
$$rU = (k + r)S_{ID}$$
$$rU = (1 + kr)S_{ID}$$

As an example, in the first scheme where $U = (k + r)S_{ID}$, the message m can be recovered from the signature (r, U) as,

$$m = r \oplus (e(U, Q_{ID})e(P, P)^{-r})).$$

The verification equations and other details of these signatures are given in Group V of Table **??**.

3.5 Embedding Previously Known ID-based Message Recovery Signatures

Recently two ID-based message recovery signature schemes have been proposed. These signatures [**?**, **?**] can be seen as special instances of our generalized scheme.

In Zhang et al.'s scheme [**?**], the signature (r, U) for the message m is computed as

$$\begin{aligned} m' &= F_1(m) \| (F_2(F_1(m)) \oplus m) \\ r &= H_2(e(P,P)^k) + m' \bmod q \\ U &= kP - rS_{ID} \end{aligned}$$

where H_2 is a secure hash function. Zhang et al.'s scheme is equivalent to MR I.3 of Table **??**, where a hash function H_2 and Abe and Okamoto's redundancy encoding is used with a slightly different computation of r.

In Tso et al.'s scheme [**?**], the signature (r, U) for the message m is computed as

$$\begin{aligned} m' &= F_1(m) \| (F_2(F_1(m)) \oplus m) \\ r &= H_2(e(P,P)^k) \oplus m' \\ U &= (k + r)S_{ID} \end{aligned}$$

where H_2 is a secure hash function. Tso et al.'s scheme is equivalent to MR IV.1 of Table **??** where a hash function H_2 and Abe and Okamoto's redundancy encoding is used.

Chapter 4

Generalized ID-Based Blind Signatures

In this chapter, we introduce the concept of generalized ID-based blind signatures. First we convert a blind ElGamal signature scheme into an ID-based counterpart. Then we generalize the signature scheme by using the ideas in Kalkan et al.'s recent work [**?**]. The generalized scheme yields many new ID-based blind signatures that have not been explored before and some of them are more efficient than the previously proposed schemes.

4.1 Background

4.1.1 Modified ElGamal Signature Scheme

Original ElGamal Signature [**?**] is not suitable to get blind signatures. However, it is possible to get blind signatures based on its variants. The modified ElGamal Signature which is used as a base tool for the rest of the paper is as follows: Let p be a large prime, q a divisor of $p-1$, and g an element in $\mathbb{Z}_p^*$ of order q. The user chooses $\alpha \in \mathbb{Z}_q$ as his private key and $\beta = g^\alpha \bmod p$ as his public key. The parameters p, q, g, and β are public whereas the user keeps α secret. To sign a

message, the user generates a random $k \in_R \mathbb{Z}_q$. Then he computes $r = g^k \bmod p$ and $s = \alpha r + km \bmod q$. The (r, s) pair is the signature of message m. The equation

$$s \equiv \alpha r + km \pmod{q} \tag{4.1}$$

is called the signature equation, and verification is done by checking the congruence $r \stackrel{?}{\equiv} (\beta^{-r} g^s)^{m^{-1}} \pmod{p}$. Security of ElGamal signatures relies on the discrete logarithm problem (DLP) since solving α from β or s from r, m, β can be reduced to solving DLP in $\mathbb{Z}_p^*$.

4.1.2 Basic Blind ElGamal Signature Scheme

Chamenish et al. [**?**] showed that the above scheme can be extended to provide blindness. The blind signature protocol in Fig. **??**. between Alice and Nancy is a blind version of the modified ElGamal signature.

Nancy		Alice
$k \in_R \mathbb{Z}_q^*$ $\tilde{r} = g^k \bmod p$		
	$\xrightarrow{\tilde{r}}$	
		$a, b \in_R \mathbb{Z}_q^*$ $r = \tilde{r}^a g^b \bmod p$ $\tilde{m} = am\tilde{r}r^{-1} \bmod q$
	$\xleftarrow{\tilde{m}}$	
$\tilde{s} = \tilde{r}\alpha + k\tilde{m}$		
	$\xrightarrow{\tilde{s}}$	
		$s = \tilde{s}r\tilde{r}^{-1} + bm \bmod q$ output (r, s)

Figure 4.1: Blind Signature Protocol

In this blind signature protocol, signature equation is $s = km + r\alpha \bmod q$ and the signature for the message m is (r, s). Verification is done by checking $r \stackrel{?}{\equiv} (\beta^{-r} g^s)^{m^{-1}} \bmod p$, which is the same as the modified ElGamal scheme. By

using the above protocol, Alice gets a valid signature for the message m from the notary (Nancy) without revealing the message.

4.1.3 Generalized Blind ElGamal Signatures

Horster et al. [**?**] showed that many variations of the basic blind signature scheme are possible by modifying the signature equation (**??**). One can use the general equation

$$\tilde{A} \equiv \alpha\tilde{B} + k\tilde{C} \pmod{q} \tag{4.2}$$

to obtain a signature, where α is the secret key of Nancy and (A, B, C) is the permutation of parameters $(\tilde{m}, \tilde{r}, \tilde{s})$. The parameter $\tilde{r}$ can be computed as $\tilde{r} = g^k$ and Alice blinds $\tilde{r}$ with two random blinding factors a, b such that $\tilde{r} = r^a g^b \bmod p$. Nancy signs the blinded message $\tilde{m}$ by using the generalized signature equation (**??**). The signature is verified by checking the equation $g^A \equiv \beta^B + r^C \pmod{p}$, where (A, B, C) is the permutation of parameters (m, r, s). In order to get a valid signature, the following two equations must hold.

$$A = a\tilde{A}C\tilde{C}^{-1} + bC \bmod q$$
$$B = b\tilde{B}C\tilde{C}^{-1} \bmod q$$

By using these equations it is possible to extract $\tilde{m}$ and s. Note that, s and $\tilde{s}$ cannot be in the equation for $\tilde{m}$ since $\tilde{m}$ is sent to Nancy before s and $\tilde{s}$ are determined in the protocol. Therefore the value s cannot appear in C. This also prevents getting a blind signature for the original ElGamal scheme.

The generalization can be extended further by choosing A, B, C as general functions of m, r, s. In that case, one of the functions should be chosen as 1 to get efficient variants. Moreover, suitable functions should be chosen to guarantee solvability of parameters $s, \tilde{s}$ and $\tilde{m}$. Further details can be found in Horster et al.'s paper [**?**].

4.2 Basic ID-based Blind Signature Scheme

An ID-based blind signature scheme consists of four algorithms: SETUP, EXTRACT, SIGN, and VERIFY. In SETUP, the PKG, chooses a secret as the global secret key and publishes the global public system parameters. In EXTRACT, the PKG verifies a user's identity and computes his private key. In SIGN, the user (Alice) and the signer (Nancy) run the blind signature protocol to get the blind signature for a message. Finally in VERIFY, the verifier verifies the signature and recovers the message by using the public parameters and the signer's identity.

An ID-based blind signature scheme can be obtained from the blind signature scheme described in Section **??** as follows:

- SETUP: Let G_1 be cyclic additive group of order q generated by P. Let G_2 be a cyclic multiplicative group of the same order and $e : G_1 \times G_1 \rightarrow G_2$ be an admissible bilinear pairing. The PKG chooses $s \in_R \mathbb{Z}_q^*$ as the global secret key and computes $P_{pub} = sP$ as the global public key. The PKG publishes system parameters $\langle G_1,\ G_2,\ e,\ P,\ P_{pub},\ H,\ H_1 \rangle$ where H and H_1 are secure hash functions.

- EXTRACT: PKG verifies the user's identity ID and computes $Q_{ID} = H_1(ID)$ and $S_{ID} = sQ_{ID}$ as user's public and private keys respectively.

- SIGN: To sign a message $m \in \mathbb{Z}_q$, Alice and Nancy run the blind signature protocol: First Nancy chooses $k \in_R \mathbb{Z}_q^*$, then computes $\tilde{r} = e(P, P)^k$ and sends $\tilde{r}$ to the Alice. After receiving $\tilde{r}$ from Nancy, Alice chooses $a, b \in_R \mathbb{Z}_q^*$, then computes $r = \tilde{r}^a e(P, P)^b$ and blinds the message m as $\tilde{m} = am\tilde{r}r^{-1}$ and sends $\tilde{m}$ to Nancy. Nancy signs the blinded message $\tilde{m}$ by using the signature equation ($\tilde{U} = \tilde{r}S_{ID} + k\tilde{m}P$) and sends $\tilde{U}$ to Alice. Alice checks whether $(\tilde{r}, \tilde{U})$ is a valid signature for $\tilde{m}$, then computes the signature U as $U = \tilde{U}r\tilde{r}^{-1} + bmP$. Finally Alice outputs the signature (r, U) for the message m. The protocol can be seen in Fig. **??**.

- VERIFY: Given ID, the message m and a signature $(\tilde{r}, \tilde{U})$, the signature is valid if the following equation holds.

$$e(U,P)e(Q_{ID},P_{pub})^{-r} = r^m$$

Nancy		Alice
$k \in_R \mathbb{Z}_q^*$ $\tilde{r} = e(P,P)^k$		
	$\xrightarrow{\tilde{r}}$	
		$a, b \in_R \mathbb{Z}_q^*$ $r = \tilde{r}^a e(P,P)^b$ $\tilde{m} = am\tilde{r}r^{-1}$
	$\xleftarrow{\tilde{m}}$	
$\tilde{U} = \tilde{r}S_{ID} + k\tilde{m}P$		
	$\xrightarrow{\tilde{U}}$	
		$e(\tilde{U},P)e(Q_{ID},P_{pub})^{-\tilde{r}} \stackrel{?}{=} \tilde{r}^{\tilde{m}}$ $U = \tilde{U}r\tilde{r}^{-1} + bmP$ output (r,U)

Figure 4.2: ID-based Blind Signature Protocol

Correctness of the given scheme can be shown by using the bilinearity properties of e. Notice that, if (r,U) is a valid signature for m, then $e(U,P)e(Q_{ID},P_{pub})^{-r}$ is

$$
\begin{aligned}
&= e(\tilde{U}r\tilde{r}^{-1} + bmP, P)e(Q_{ID}, P_{pub})^{-r} \\
&= e(\tilde{U}r\tilde{r}^{-1} + bmP, P)e(-rS_{ID}, P) \\
&= e((\tilde{r}S_{ID} + k\tilde{m}P)r\tilde{r}^{-1} + bmP, P)e(-rS_{ID}, P) \\
&= e(rS_{ID} + k\tilde{m}r\tilde{r}^{-1}P + bmP, P)e(-rS_{ID}, P) \\
&= e(k\tilde{m}r\tilde{r}^{-1}P + bmP, P) \\
&= e(k(am\tilde{r}r^{-1})r\tilde{r}^{-1}P + bmP, P) \\
&= e(kamP + bmP, P) \\
&= (\tilde{r}^a + e(P, P)^b)^m \\
&= r^m
\end{aligned}
$$

The above scheme is the ID-based version of the modified blind ElGamal signature described in Section **??**. In that scheme, the signature equation is $\tilde{s} = \alpha\tilde{r} + k\tilde{m} \bmod q$ where $\tilde{r} = g^k$ and the signature is (r, s). Since additive elliptic curve groups are used in the ID-based structure, the signing equation and $\tilde{r}$ are slightly different. The signing equation for the ID-based ElGamal signature becomes,

$$\tilde{U} = \tilde{r}S_{ID} + k\tilde{m}P$$

In this signature equation, uppercase letters are used to denote the elements of the elliptic curve group. S_{ID} is the private key of the user; so it is a natural replacement for α in the original scheme. U is the second part of the signature, replacing s. A natural choice for $\tilde{r}$ in the ID-based scheme is $\tilde{r} = e(P, P)^k$ since $\tilde{r} = g^k$ in the original scheme.

4.2.1 Blindness Proof

A signature is said to be blind if a given message-signature pair and Nancy's view are statistically independent. That is, the signer cannot get any information on the actual message and the resulting signature. If there always exists a unique mapping between any view of the signer and any given message signature pair, we can say that the signature is blind.

In order to prove blindness we will show that for a given message-signature pair (m, r, U) and any view of Nancy $(\tilde{m}, \tilde{r}, \tilde{U})$, there always exists a unique pair of blinding factors a, b that maps $(\tilde{m}, \tilde{r}, \tilde{U})$ to (m, r, U). Since Alice chooses a, b randomly, Nancy cannot get any information from her view and the signature scheme will be blind.

For a signature (r, U) generated for message m during the protocol, the following equations must hold.

$$\tilde{m} = am\tilde{r}r^{-1} \tag{4.3}$$

$$r = \tilde{r}^a e(P, P)^b \tag{4.4}$$

$$U = \tilde{U}r\tilde{r}^{-1} + bmP \tag{4.5}$$

The blinding factors a and b can be uniquely determined from the first two equations. a is determined uniquely from (**??**) as $a = \tilde{m}m^{-1}r\tilde{r}^{-1}$. From (**??**), $e(P, P)^b = r\tilde{r}^{-a}$, since $e(P, P)$ is a generator for G_2, therefore b is also unique. If these a and b satisfy (**??**), the desired mapping will be found and the signature will be blind. We know that

$$U = \tilde{U}r\tilde{r}^{-1} + bmP \Longleftrightarrow e(U, P) = e(\tilde{U}r\tilde{r}^{-1} + bmP, P).$$

So it is sufficient to show that $e(U, P) = e(\tilde{U}r\tilde{r}^{-1} + bmP, P)$ to complete the proof. Notice that, since (r, U) is a valid signature, the signature equation $\tilde{U} = \tilde{r}S_{ID} + k\tilde{m}P$ and the verification equation $e(U, P)e(Q_{ID}, P)^{-r} = r^m$ should hold.

No.	$\tilde{r}$	$\tilde{U}$	r	U	$\tilde{m}$	Verification
BL I.1	$e(P,P)^k$	$\tilde{r}S_{ID} + k\tilde{m}P$	$\tilde{r}^a e(P,P)^b$	$\tilde{U}r\tilde{r}^{-1} + bmP$	$am\tilde{r}r^{-1}$	$e(U,P)e(Q_{ID}, P_{pub})^{-r} = r^m$
BL I.2	$e(P,P)^k$	$\tilde{m}S_{ID} + k\tilde{r}P$	$\tilde{r}^a e(P,P)^b$	$ar\tilde{r}^{-1}\tilde{U} + brP$	$a^{-1}m\tilde{r}r^{-1}$	$e(U,P)e(Q_{ID}, P_{pub})^{-m} = r^r$
BL II.1	$e(P,P)^k$	$S_{ID} + k\tilde{m}\tilde{r}P$	$\tilde{r}^a e(P,P)^b$	$\tilde{U} + bmrP$	$amr\tilde{r}^{-1}$	$e(U,P)e(Q_{ID}, P_{pub}) = r^{mr}$
BL II.2	$e(P,P)^k$	$\tilde{m}\tilde{r}S_{ID} + kP$	$\tilde{r}^a e(P,P)^b$	$a\tilde{U} + bP$	$a^{-1}mr\tilde{r}^{-1}$	$e(U,P)e(Q_{ID}, P_{pub})^{-mr} = r$
BL III.1	$a^{-1}r$	$\tilde{r}S_{ID} + kP$	$H(m,t)$	$a\tilde{U} + bP$	–	$H(m, e(U,P)e(Q_{ID}, P_{pub})^{-r}) = r$
BL III.2	ar	$-S_{ID} + k\tilde{r}P$	$H(m,t)$	$\tilde{U} + brP$	–	$H(m, e(U,P)^{r^{-1}} e(Q_{ID}, P_{pub})^{r^{-1}}) = r$

Table 4.1: Generalized ID-based blind signatures, where $\tilde{t} = e(P,P)^k$ and $t = \tilde{t}^a e(P,P)^b$

Hence, we have $e(\tilde{U}r\tilde{r}^{-1} + bmP, P)$ equals

$$\begin{aligned}
&= e(\tilde{U}r\tilde{r}^{-1}, P)e(P,P)^{bm} \\
&= e(\tilde{U}r\tilde{r}^{-1,P})(r\tilde{r}^{-a})^m \\
&= e(\tilde{U}, P)^{r\tilde{r}^{-1}} r^m \tilde{r}^{-am} \\
&= e(\tilde{r}S_{ID} + k\tilde{m}P, P)^{r\tilde{r}^{-1}} r^m \tilde{r}^{-am} \\
&= e(\tilde{r}S_{ID}, P))^{r\tilde{r}^{-1}} e(k\tilde{m}P, P)^{r\tilde{r}^{-1}} r^m \tilde{r}^{-am} \\
&= e(S_{ID}, P)^r \tilde{r}^{\tilde{m}r\tilde{r}^{-1}} r^m \tilde{r}^{-am} \\
&= e(Q_{ID}, P)^r \tilde{r}^{am} r^m \tilde{r}^{-am} \\
&= e(Q_{ID}, P)^r r^m \\
&= e(U, P).
\end{aligned}$$

4.3 Generalized ID-based Blind Signatures

We can generalize the above signature scheme by using different signature equations. Instead of using $\tilde{U} = \tilde{r}S_{ID} + k\tilde{m}P$ as the signature equation, we can use

$$A = BS_{ID} + kC$$

in general, where (A, B, C) is the permutation of the parameters $(\tilde{m}, \tilde{r}, \tilde{U})$. Note that, we can use P, $\tilde{m}P$ and $\tilde{r}P$ instead of 1, $\tilde{m}$ and $\tilde{r}$ in cases where they need to be members of the elliptic curve group. However, not all the permutations generate useful variants. We should consider that $\tilde{U}$ is a member of the elliptic curve group so it cannot be used for B. Moreover, $\tilde{U}$ cannot be in the position of C. Since, in that case, U and $\tilde{U}$ are needed to extract $\tilde{m}$; but, $\tilde{m}$ is sent to Nancy

before U and $\tilde{U}$ are determined in the protocol. Therefore we can get only two variants. The signing equation for these variants are:

$$\tilde{U} = \tilde{m}S_{ID} + k\tilde{r}P \tag{4.6}$$

$$\tilde{U} = \tilde{r}S_{ID} + k\tilde{m}P \tag{4.7}$$

Two more variants can be generated by using the permutations of $(\tilde{m}\tilde{r}, \tilde{U}, 1)$. The signing equation for these variants are:

$$\tilde{U} = S_{ID} + k\tilde{m}\tilde{r}P \tag{4.8}$$

$$\tilde{U} = \tilde{m}\tilde{r}S_{ID} + kP \tag{4.9}$$

The verification equations and other details for these signatures are summarized in Table **??**. Note that, we can also use a general function $f(\tilde{m}, \tilde{r})$ instead of just the product $\tilde{m}\tilde{r}$.

Another way of using ElGamal signatures to sign a message m is to mix m into r by a hash function, instead of using m in the computation of U. In this way, it is possible to remove $\tilde{m}$ from the signing equations by modifying the blind signature protocol. If we remove $\tilde{m}$ from (**??**), the signing equation will be,

$$\tilde{U} = \tilde{r}S_{ID} + kP. \tag{4.10}$$

If we use (**??**) as the signature equation, we modify the blind signature protocol as follows: Instead of sending $\tilde{r}$, Nancy computes $\tilde{t} = e(P, P)^k$ and sends $\tilde{t}$ to Alice. Alice computes $t = \tilde{t}^a e(P, P)^k$ and $r = H(m, t)$, where H is a secure hash function. Then, she computes $\tilde{r} = a^{-1}r$ and sends $\tilde{r}$ to Nancy. Nancy computes $\tilde{U}$ by using the signature equation (**??**) and sends $\tilde{U}$ to Alice. Alice checks whether the signature is valid, computes $U = a\tilde{U} + bP$, and outputs the signature (r, U). The modified protocol can be found in Fig. **??**.

Similarly, if we remove $\tilde{m}$ from (**??**) the signing equation will be,

$$\tilde{U} = S_{ID} + k\tilde{r}P.$$

The verification equation and other details for these signatures can be found in Table **??**. Note that, removing $\tilde{m}$ from (**??**) and (**??**) does not generate new variants.

Nancy		Alice
$k \in_R \mathbb{Z}_q^*$ $\tilde{t} = e(P,P)^k$		
	$\xrightarrow{\tilde{t}}$	
		$a, b \in_R \mathbb{Z}_q^*$ $t = \tilde{t}^a e(P,P)^b$ $r = H(m,t)$ $\tilde{r} = a^{-1}r$
	$\xleftarrow{\tilde{r}}$	
$\tilde{U} = \tilde{r}S_{ID} + kP$		
	$\xrightarrow{\tilde{U}}$	
		$e(\tilde{U}, P)e(Q_{ID}, P_{pub})^{-\tilde{r}} \stackrel{?}{=} \tilde{r}$ $U = a\tilde{U} + bP$ output (r, U)

Figure 4.3: Modified Blind Signature Protocol

4.4 More Efficient ID-based Blind Signatures

Computing a signature requires two to four scalar multiplications in G_1 and three or four exponentiations in G_2, depending on the signature equation, as well as one pairing evaluation. The other pairing $e(Q_{ID}, P_{pub})$ can be precomputed before the signature protocol.

The cost of verifying a signature will be dominated by the pairing computations, which is the most expensive operation. Two pairing computations and an exponentiation in G_1 are needed to verify a signature. Note that, in the proposed schemes, the value $e(Q_{ID}, P_{pub})$ is used, which is fixed for a particular user and needs to be computed only once for each user.

The number of pairing operations can be reduced to one by changing the definitions of S_{ID} and Q_{ID} as in [**?**]. If we define

$$Q_{ID} = (H_1(ID) + s)P$$
$$S_{ID} = (H_1(ID) + s)^{-1}P,$$

No.	$\tilde{r}$	$\tilde{U}$	r	U	$\tilde{m}$	Verification
BL IV.1	$e(P,Q_{ID})^k$	$\tilde{r}S_{ID}+k\tilde{m}P$	$\tilde{r}^a e(P,Q_{ID})^b$	$\tilde{U}r\tilde{r}^{-1}+bmP$	$am\tilde{r}r^{-1}$	$e(U,Q_{ID})e(P,P)^{-r}=r^m$
BL IV.2	$e(P,Q_{ID})^k$	$\tilde{m}S_{ID}+k\tilde{r}P$	$\tilde{r}^a e(P,Q_{ID})^b$	$ar\tilde{r}^{-1}\tilde{U}+brP$	$a^{-1}m\tilde{r}r^{-1}$	$e(U,Q_{ID})e(P,P)^{-m}=r^r$
BL IV.3	$e(P,Q_{ID})^k$	$S_{ID}+k\tilde{m}\tilde{r}P$	$\tilde{r}^a e(P,Q_{ID})^b$	$\tilde{U}+bmrP$	$amr\tilde{r}^{-1}$	$e(U,Q_{ID})e(P,P)=r^{mr}$
BL IV.4	$e(P,Q_{ID})^k$	$\tilde{m}\tilde{r}S_{ID}+kP$	$\tilde{r}^a e(P,Q_{ID})^b$	$a\tilde{U}+bP$	$a^{-1}mr\tilde{r}^{-1}$	$e(U,Q_{ID})e(P,P)^{-mr}=r$
BL IV.5	$a^{-1}r$	$\tilde{r}S_{ID}+kP$	$H(m,t)$	$a\tilde{U}+bP$	–	$H(m,e(U,Q_{ID})e(P,P)^{-r})=r$
BL IV.6	ar	$-S_{ID}+k\tilde{r}P$	$H(m,t)$	$\tilde{U}+brP$	–	$H(m,e(U,Q_{ID})^{r^{-1}}e(P,P)^{r^{-1}})=r$
BL V.1	$e(P,P)^k$	$(\tilde{r}+k\tilde{m})S_{ID}$	$\tilde{r}^a e(P,Q_{ID})^b$	$\tilde{U}r\tilde{r}^{-1}+bmP$	$am\tilde{r}r^{-1}$	$e(U,Q_{ID})e(P,P)^{-r}=r^m$
BL V.2	$e(P,P)^k$	$(k+\tilde{r}\tilde{m})S_{ID}$	$\tilde{r}^a e(P,Q_{ID})^b$	$ar\tilde{r}^{-1}\tilde{U}+brP$	$a^{-1}m\tilde{r}r^{-1}$	$e(U,Q_{ID})e(P,P)^{-m}=r^r$
BL V.3	$e(P,P)^k$	$(\tilde{m}+k\tilde{r})S_{ID}$	$\tilde{r}^a e(P,Q_{ID})^b$	$\tilde{U}+bmrP$	$amr\tilde{r}^{-1}$	$e(U,Q_{ID})e(P,P)=r^{mr}$
BL V.4	$e(P,P)^k$	$(1+k\tilde{m}\tilde{r})S_{ID}$	$\tilde{r}^a e(P,Q_{ID})^b$	$a\tilde{U}+bP$	$a^{-1}mr\tilde{r}^{-1}$	$e(U,Q_{ID})e(P,P)^{-mr}=r$
BL V.5	$a^{-1}r$	$(\tilde{r}+k)S_{ID}$	$H(m,t)$	$a\tilde{U}+bP$	–	$H(m,e(U,Q_{ID})e(P,P)^{-r})=r$
BL V.6	ar	$(1+k\tilde{r})S_{ID}$	$H(m,t)$	$\tilde{U}+brP$	–	$H(m,e(U,Q_{ID})^{r^{-1}}e(P,P)^{r^{-1}})=r$

Table 4.2: Generalized ID-based blind signatures, where $\tilde{t}=e(P,Q_{ID})^k$ in IV.5, IV.6, $\tilde{t}=e(P,P)^k$ in V.5, V.6 and $t=\tilde{t}^a e(P,Q_{ID})^b$

the number of pairing evaluations can be reduced to one. Note that Q_{ID} can be computed by anyone, since the value of sP is public, but S_{ID} cannot be computed without knowing the value of s.

By changing the definitions of S_{ID} and Q_{ID} as described, we can get more efficient variants of the proposed schemes. The computation of r should also be changed in order to adapt to the changes. Instead of computing $r=e(P,P)^k$, we have

$$r=e(P,Q_{ID})^k.$$

This modification does not affect the efficiency of the signature computation, since the value $e(P,Q_{ID})$ can be precomputed by the sender.

The verification equations and other details of the efficient versions of the signatures modified in this fashion are given in Group IV of Table **??**.

Further variants with a reduced signing cost can be obtained by modifying the generalized signature equation as,

$$U=AS_{ID}+kBS_{ID},$$

where the signing cost is reduced by one scalar multiplication in the elliptic curve group G_1. Note that, this kind of generalization is not possible over the basic ElGamal signatures, because when k and α are used together, we cannot extract s from the signing equation.

We can get six more efficient variants by this modification whose signing equations are:

$$U = (r + km)S_{ID}$$
$$U = (k + rm)S_{ID}$$
$$U = (m + kr)S_{ID}$$
$$U = (1 + kmr)S_{ID}$$
$$U = (r + k)S_{ID}$$
$$U = (1 + kr)S_{ID}$$

The verification equations and other details of these signatures are given in Group V of Table **??**.

4.5 Performance Comparison

In this section, we give a performance comparison of our proposed schemes and the four available ID-based blind signature schemes [**?**, **?**, **?**, **?**] based on bilinear pairings. As the main computational cost, we consider the number of bilinear pairings (denoted by B), modular exponentiations, (denoted by E), and scalar multiplications in elliptic curve group (denoted by M). We assume the value of $e(P, P)$ is precomputed by every party, and the value of $e(P, Q_{ID})$ is precomputed by the signer but not the verifier.

Among the proposed schemes, Group I, Group II, and Group III are the least efficient schemes with signing cost of one pairing, two to four scalar multiplications, and three or four exponentiations and verification cost of two pairings, and one or two exponentiations. Group IV and Group V are the most efficient schemes with the signing cost of one pairing, two to four scalar multiplications, and three or four exponentiations and verification cost of one pairing, and one or two exponentiations.

Compared to the previously proposed schemes, ZK02 [**?**] has the signing cost of $2B+6M$ and verification cost of $2B+1E$. In ZK03 [**?**], signing cost is $2B+6M$

Scheme	Signing Cost	Verification Cost
Group I	$1B + 4M + 4E$	$2B + 2E$
Group II	$1B + (2\text{-}4)M + 3E$	$2B + 1E$
Group III	$1B + (2\text{-}4)M + 3E$	$2B + 1E$
Group IV	$1B + (2\text{-}4)M + (3\text{-}4)E$	$1B + (1\text{-}2)E$
Group V	$1B + (2\text{-}3)M + (3\text{-}4)E$	$1B + (1\text{-}2)E$
ZK02 [?]	$2B + 6M$	$2B + 1E$
ZK03 [?]	$2B + 6M$	$2B + 1M$
HCW05 [?]	$1B + 3M + 3E$	$2B + 1M$
GWWL07 [?]	$3B + 7M$	$4B$

Table 4.3: Comparison of ID-Based Blind Signature Schemes

and verification cost is $2B + 1M$. In HCW05 [?], signing cost is $1B + 3M + 3E$ and verification cost is $2B + 1M$. In GWWL07 [?] signing cost is $3B + 7M$ and verification cost is $4B$; however, GWWL07 [?] has the advantage that blind signature protocol needs only one round.

Performance comparison of our schemes to the previously proposed schemes can be found in Table ??. As the table shows Group IV and Group V are the most efficient signatures with the smallest number of pairing evaluations.

Chapter 5

Generalized ID-Based Signcryption Schemes

In this chapter, we introduce the concept of generalized ID-based signcryption scheme. We show how an ID-based signature scheme can be converted into an ID-based signcryption scheme and describe the details of the basic scheme. Then we generalize the basic scheme. The generalized scheme yields many new signcryption schemes that have not been explored before.

5.1 Model of ID-based Signcryption Scheme

An ID-based signcryption scheme consists of four algorithms:

- SETUP: The private key generator (PKG), a trusted authority, chooses the global secret key, computes the global public key and publishes it with other system parameters.

- EXTRACT: PKG verifies the user's identity and computes user's public and private key. The private key should be sent to the user over a secure channel

after this phase.

- SIGN/ENCRYPT: An algorithm that takes the message m, sender's private key, recipient's public key, and other public system parameters as input; produces a signature on m using sender's private key; produces a mask of m using recipient's public key; and outputs a ciphertext that includes the signature and the mask.

- DECRYPT/VERIFY: An algorithm that takes receiver's private key, a ciphertext, and other public system parameters as input; removes the mask on the message using receiver's private key; and outputs 1 if the signature is valid. Otherwise it outputs 0.

5.2 Basic ID-based Signcryption Scheme

An ID-based signature can be converted into an ID-based signcryption scheme as follows: Instead of sending the message m with the signature, m is masked using the recipient's public key. The mask should be removed before checking that the signature is valid. Recipient uses his private key to remove the mask.

The ID-based signcryption scheme can be obtained from ID I.1 of Table **??** as follows:

- SETUP: Let G_1 be a cyclic additive group of order q generated by P. Let G_2 be a cyclic multiplicative group of the same order and $e : G_1 \times G_1 \rightarrow G_2$ be an admissible bilinear pairing. The PKG chooses $s \in_R \mathbb{Z}_q^*$ as the global secret key and computes $P_{pub} = sP$ as the global public key. The PKG publishes system parameters $\langle G_1,\ G_2,\ e,\ P,\ P_{pub},\ H_1,\ H_2,\ H_3 \rangle$ where H_1, H_2, and H_3 are secure hash functions.

- EXTRACT: PKG verifies the user's identity ID and computes $Q_{ID} = H_1(ID)$ and $S_{ID} = sQ_{ID}$ as user's public and private keys respectively.

- SIGN/ENCRYPT: To sign a message $m \in \mathbb{Z}_q$, a sender with his private key S_{ID_A}, first chooses $k \in_R \mathbb{Z}_q$, then masks the message m by using recipient's public key as,

$$c = m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$$

then computes,

$$\begin{aligned} V &= kP \\ r &= H_2(V) \\ U &= k^{-1}(cP - rS_{ID_A}) \end{aligned}$$

and the ciphertext for m is (c, U, V)

- DECRYPT/VERIFY: Given ID, a message m, and a ciphertext (c, U, V), first the receiver removes the mask as,

$$m = c \oplus H_3(e(S_{ID_B}, V)), \tag{5.1}$$

and computes $r = H(V)$. The signature is valid if the following equation holds.

$$e(U, V)e(Q_{ID}, P_{pub})^r \stackrel{?}{=} e(P, P)^c$$

Correctness of the given scheme can be shown by using the bilinearity properties of e. If (c, U, V) is a valid ciphertext, then the mask successfully removed on m since

$$\begin{aligned} e(S_{ID_B}, V) &= e(S_{ID_B}, kP) \\ &= e(Q_{ID_B}, kP_{pub}) \\ &= e(Q_{ID_B}, P_{pub})^k \end{aligned}$$

If (V,U) is a valid signature for m then we have:

$$\begin{aligned} e(U,V)e(Q_{ID},P_{pub})^r &= e(k^{-1}(cP - rS_{ID}),kP)e(Q_{ID},P_{pub})^r \\ &= e(cP - rS_{ID},P)e(rS_{ID},P) \\ &= e(cP,P) \\ &= e(P,P)^c \end{aligned}$$

As pointed out in [**?**], if the signature equation depends on only the public parameters and the message, then the resulting ciphertext is not secure. This is because, if a user signcrypts two messages m_1, m_2 and outputs a ciphertext of one of the messages, then an attacker can verify the signature on the message by simply trying m_1 and m_2 one by one in the verification equation and find out which message matches the ciphertext. Therefore, we use the mask c instead of m in the signing equations. In that case, an attacker can verify a signature by using the public parameters and the ciphertext. However, he cannot get any information on the message since m is not used in signature verification..

5.3 The Generalized ID-Based Signcryption Scheme

We can generalize the above signcryption scheme. All signatures in Table **??** except Group VI can be converted into signcryption schemes. The idea is masking the message using the recipient's public key and the receiver removes the mask with his private key before checking that the signature is valid. Notice that in some signatures in Table **??**, r is computed as $r = e(P,P)^k$ in order to reduce the cost of verification by one pairing computation. However, r cannot be computed as $e(P,P)^k$ in the signcryption case since the value of kP is needed to remove the mask on the message. Therefore, r is computed as $H(kP)$ for those signcryption schemes. The verification equations and other details for these signcryption schemes are summarized in Table **??**.

Signatures in Group VI of Table **??** cannot be converted into signcryption

No.	r	U	c	Verification
SC I.1	$H(V)$	$k^{-1}(cP - rS_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^r = e(P, P)^c$
SC I.2	$H(V)$	$k^{-1}(rP - cS_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^c = e(P, P)^r$
SC I.3	$H(V)$	$kcP - rS_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^r = e(P, kP)^c$
SC I.4	$H(V)$	$rkP - cS_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^c = e(P, kP)^r$
SC I.5	$H(c, V)$	$k^{-1}(P - rS_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^r = e(P, P)$
SC I.6	$H(c, V)$	$k^{-1}(rP - S_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub}) = e(P, P)^r$
SC I.7	$H(c, V)$	$kP - rS_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^r = e(P, kP)$
SC I.8	$H(c, V)$	$rkP - S_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub}) = e(P, kP)^r$
SC II.1	$H(V)$	$k^{-1}(P - crS_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^{cr} = e(P, P)$
SC II.2	$H(V)$	$k^{-1}(-S_{ID} + crP)$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub}) = e(P, P)^{cr}$
SC II.3	$H(V)$	$kP - crS_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^{cr} = e(P, kP)$
SC II.4	$H(V)$	$crkP - S_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub}) = e(P, kP)^{cr}$
SC III.1	$H(V)$	$k^{-1}(c^{-1}P - rS_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^r = e(P, P)^{c^{-1}}$
SC III.2	$H(V)$	$k^{-1}(rP - c^{-1}S_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^{c^{-1}} = e(P, P)^r$
SC III.3	$H(V)$	$c^{-1}kP - rS_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^r = e(P, kP)^{c^{-1}}$
SC III.4	$H(V)$	$rkP - c^{-1}S_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^{c^{-1}} = e(P, kP)^r$
SC IV.1	$H(V)$	$k^{-1}(cP - r^{-1}S_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^{r^{-1}} = e(P, P)^c$
SC IV.2	$H(V)$	$k^{-1}(r^{-1}P - cS_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^c = e(P, P)^{r^{-1}}$
SC IV.3	$H(V)$	$ckP - r^{-1}S_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^{r^{-1}} = e(P, kP)^c$
SC IV.4	$H(V)$	$r^{-1}kP - cS_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^c = e(P, kP)^{r^{-1}}$
SC IV.5	$H(c, V)$	$k^{-1}(P - r^{-1}S_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub})^{r^{-1}} = e(P, P)$
SC IV.6	$H(c, V)$	$k^{-1}(r^{-1}P - S_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)e(Q_{ID}, P_{pub}) = e(P, P)^{r^{-1}}$
SC IV.7	$H(c, V)$	$kP - r^{-1}S_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub})^{r^{-1}} = e(P, kP)$
SC IV.8	$H(c, V)$	$r^{-1}kP - S_{ID}$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, P)e(Q_{ID}, P_{pub}) = e(P, kP)^{r^{-1}}$
SC V.1	$H(V)$	$k^{-1}r^{-1}(cP - S_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)^r e(Q_{ID}, P_{pub}) = e(P, P)^c$
SC V.2	$H(V)$	$k^{-1}r^{-1}(P - cS_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)^r e(Q_{ID}, P_{pub})^c = e(P, P)$
SC V.3	$H(c, V)$	$k^{-1}r^{-1}(P - S_{ID})$	$m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$	$e(U, kP)^r e(Q_{ID}, P_{pub}) = e(P, P)$

Table 5.1: The generalized ID-based signcryption scheme and their verification equations, where $V = kP$ and the ciphertext is (c, U, V).

schemes. The reason behind is that kQ_{ID} is sent to the recipient in these signatures and without kP recipient cannot remove the mask on the message. Fortunately, this problem can be solved by changing the definitions of public and private keys. This modification yields eight more signcryption schemes that includes the most efficient signcryption scheme proposed by Barreto et al. [**?**].

If we define

$$Q_{ID} = (H_1(ID) + s)P$$
$$S_{ID} = (H_1(ID) + s)^{-1}P,$$

then the number of pairing evaluations can be reduced to one. Note that Q_{ID} can be computed by anyone, since the value of sP is public, but S_{ID} cannot be computed without knowing the value of s.

By changing the definitions of S_{ID} and Q_{ID} as described, we can convert the signatures group VI of Table **??**. The computation of the mask should also be changed in order to adapt to the changes. Instead of computing $c = m \oplus H_3(e(Q_{ID_B}, P_{pub})^k)$, the mask will be computed as

$$c = m \oplus H_3(e(P, P)^k).$$

, and kQ_{ID_B} is sent to the recipient instead of kP since recipient needs the value of kQ_{ID_B} to remove the mask. The computation of r should also be changed since the ciphertext does not include kP anymore. r will be computed as $r = e(P, P)^k$ or $r = H(m, e(P, P)^k)$ depending on the signing equation.

As an example, consider the modified version of VI.1 of Table **??** where $U = (r + km)S_{ID_A}$. The value of r will be computed as $r = e(P, P)^k$, the mask will be computed as $c = m \oplus H_3(e(P, P)^k)$, and $V = kQ_{ID_B}$. The ciphertext will be (c, U, V). To verify the signature, the recipient computes the following:

$$e(P, P)^k = e(V, S_{ID_B})$$
$$c = m \oplus H_3(e(P, P)^k).$$
$$e(U, Q_{ID_A})e(P, P)^{-r} \stackrel{?}{=} e(P, P)^{km} \tag{5.2}$$

The signature is valid if equation **??** holds.

No.	r	U	c	Verification
SC VI.1	$H(V)$	$(r+km)S_{ID_A}$	$m \oplus H_3(e(P,P)^k)$	$e(U, Q_{ID_A})e(P,P)^{-r} = e(P,P)^{km}$
SC VI.2	$H(V)$	$(m+kr)S_{ID_A}$	$m \oplus H_3(e(P,P)^k)$	$e(U, Q_{ID_A})e(P,P)^{-m} = e(P,P)^{kr}$
SC VI.3	$H(V)$	$(rm+k)S_{ID_A}$	$m \oplus H_3(e(P,P)^k)$	$e(U, Q_{ID_A})e(P,P)^{-rm} = e(P,P)^{k}$
SC VI.4	$H(V)$	$(1+kmr)S_{ID_A}$	$m \oplus H_3(e(P,P)^k)$	$e(U, Q_{ID_A})e(P,P)^{-1} = e(P,P)^{kmr}$
SC VI.5	$H(V)$	$r^{-1}(m+k)S_{ID_A}$	$m \oplus H_3(e(P,P)^k)$	$e(U, Q_{ID_A})^r e(P,P)^{-m} = e(P,P)^{k}$
SC VI.6	$H(V)$	$r^{-1}(1+kmS_{ID_A})$	$m \oplus H_3(e(P,P)^k)$	$e(U, Q_{ID_A})^r e(P,P)^{-1} = e(P,P)^{km}$
SC VI.7	$H(m,V)$	$(r+k)S_{ID_A}$	$m \oplus H_3(e(P,P)^k)$	$e(U, Q_{ID_A})e(P,P)^{-}r = e(P,P)^{k}$
SC VI.8	$H(m,V)$	$r^{-1}(1+k)S_{ID_A}$	$m \oplus H_3(e(P,P)^k)$	$e(U, Q_{ID_A})^r e(P,P)^{-}1 = e(P,P)^{k}$

Table 5.2: The modified ID-based signcryption schemes and their verification equations, where $V = kQ_{ID_B}$ and the ciphertext is (c, U, V).

To verify a signature, a user should know the value of $e(P,P)^k$, but he cannot find it without S_{ID_B}. So, we do not need to use c instead of m as in the case of the signcryption schemes in table **??**.

The verification equations and other details for modified signcryption schemes are summarized in Table **??**.

5.4 Efficiency of the Proposed Schemes

As the main computational cost, we consider the number of bilinear pairings, modular exponentiations, and scalar multiplications in the elliptic curve group. We assume the value of $e(P,P)$ is precomputed by every party.

In the signcryption schemes in Table **??**, computing a ciphertext (the signature and the mask) requires two or three scalar multiplications in G_1, one exponentiation in G_2, and a pairing evaluation. Note that the value $e(Q_{ID_B}, P_{pub})$ is fixed for a particular user, so it needs to be computed once for each user.

The cost of removing the mask and verifying the signature will be dominated by pairing computations, which is the most expensive operation. Removing a mask requires one pairing computation and verifying a signature requires two or three pairing computations depending on the signature equation. Note that, the value $e(P,P)$ is fixed, so it needs to be completed only once. Also the value $e(Q_{ID_A}, P_{pub})$ needs to be computed only once for each user.

In the modified signcryption schemes in Table **??**, computing a ciphertext requires two scalar multiplications in G_1, one exponentiation in G_2. Note that, computing a ciphertext does not need a pairing evaluation since $e(P, P)$ is a fixed value.

The cost of removing the mask and verifying the signature require only two pairing evaluations and two exponentiations in G_2. These modified schemes are far more efficient than the previously proposed schemes.

5.5 Embedding Previously Known ID-based Signcryption Schemes

Recently several ID-based signcryption schemes have been proposed. Two of these schemes [**?**, **?**] can be seen as special instances of our generalized scheme, three of them use same signatures equations as in our generalized scheme.

In Malone-Lee's scheme [**?**], the ciphertext (c, U, V) for the message m is computed as

$$\begin{aligned} V &= kP \\ r &= H_2(V, m) \\ U &= rS_{ID_A} + kP_{pub} \\ c &= m \oplus H_3(e(Q_{ID_B}, kP_{pub})) \end{aligned}$$

Malone-Lee's scheme is equivalent to SC I.7 of Table **??** where m is used instead of c in the computation of r and a slightly different signing equation is used.

In Barreto et al.'s scheme [**?**], the ciphertext (c, U, V) for the message m is

computed as

$$
\begin{aligned}
V &= kQ_{ID_B} \\
r &= H_2(V, m) \\
U &= (k + r)S_{ID_A} \\
c &= m \oplus H_3(e(P, P)^k)
\end{aligned}
$$

Barreto et al.'s scheme is same as SC VI.7 of Table **??**.

In Libert et al.'s [**?**] scheme, McCullagh and Barreto's [**?**] scheme, and Malone-Lee's [**?**] improved scheme, the different masking techniques are used, however, the signature equations are same as the signature equations of SC I.7, SC VI.7, SC VI.7 respectively.

Chapter 6

Conclusion and Future Work

The ElGamal signature scheme is a key tool for constructing ID-based signature schemes. We used the ElGamal signature to obtain an ID-based signature scheme. Then we showed how the basic ID-based ElGamal signature scheme can be extended into a generalized ID-based signature scheme as in the work of Horster et al. on basic ElGamal signatures [**?**]. We discussed which variants are not possible and which variants are not secure in the ID-based setting. We also presented some original variants which were not possible on the basic ElGamal scheme.

Most of the ID-based signatures in the literature [**?**, **?**, **?**, **?**] can be seen as special instances of the generalized ID-based signature scheme described in this paper. Therefore, our generalized scheme provides a unified framework for many of the previously proposed ID-based signatures. This framework also yields many new ID-based signature schemes that have not been explored before.

We extended our work to provide additional properties to our signature schemes. We investigated ID-based signatures giving message recovery. We showed how the basic ElGamal signature with message recovery can be converted to an ID-based signature with message recovery. Then again we extended our ID-based signature scheme into a generalized ID-based message recovery signature as in the work of Horster et al. [**?**] on basic ElGamal signatures with message

recovery. We also presented some original variants which were not possible in the non-ID-based setting. Then, we modified some of our signatures to get more efficient signatures providing message recovery.

The two ID-based message recovery signatures in the literature [**?**, **?**] can be seen as special instances of the generalized scheme. Our work also yields many new ID-based signatures with message recovery that have not been explored before.

Among the proposed schemes with message recovery Group IV and Group V of Table **??** are the most efficient signatures, with just one pairing operation needed in signature verification. Group V has the further advantage of reducing the cost of the signature operation by one scalar multiplication in the elliptic curve group G_1.

ID-based blind signatures are also investigated in this thesis. We showed how a modified blind ElGamal signature can be converted to an ID-based blind signature. We extended our basic ID-based blind signature scheme into a generalized ID-based blind signature as in the work of Horster et al. [**?**] on the basic blind ElGamal signature. We also presented some original variants which were not possible in the non-ID-based setting. Then, we modified some of our signatures to get more efficient blind signatures.

Among the proposed blind signatures, Group IV and Group V of Table **??** with just one pairing operation in signature verification, become the most efficient ID-based blind signatures in the literature.

Lastly, ID-based blind signcryption schemes are investigated. We showed how an ID-based signature scheme can be converted into an ID-based signcryption scheme. Then we generalized the idea and obtained signcryption schemes from our ID-based signatures. Our work yields many new ID-based signcryption schemes that have not been explored before.

For future work, ways of proving the security of the proposed ID-based signature schemes can be investigated. One can also try to improve the efficiency of

the proposed signature schemes by changing the signature and verification equations. The ideas presented in this thesis can also be used to get new ID-based signatures with more additional features. One can also try to use the ideas in this thesis to get generalized versions of ID-based group signatures or hierarchical ID-based signatures.

Bibliography

[1] M. Abe and T. Okamoto. A signature scheme with message recovery as secure as discrete logarithm. In *Proc. of ASIACRYPT'99*, volume 1716 of *LNCS*, pages 378–389. Springer-Verlag, 1999.

[2] P. S. L. M. Barreto, B. Libert, N.McCullagh, and J. Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Proc. of ASIACRYPT'05*, volume 3778 of *LNCS*, pages 515–532, 2005.

[3] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Proc. of CRYPTO'01*, volume 2139 of *LNCS*, pages 213–229. Springer-Verlag, 2001.

[4] X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In *In Advances in Cryptology - CRYPTO 2003*, volume 2729 of *LNCS*, pages 382–398. Springer-Verlag, 2003.

[5] J. L. Camenisch, J.-M. Piveteau, and M. A. Stadler. Blind signatures based on the discrete logarithm problem. In *Proc. of Eurocrypt 1994*, volume 950 of *LNCS*, pages 428–432, 1995.

[6] J. Cha and J. Cheon. An identity-based signature from gap diffie-hellman group. In *Proc. of PKC 2003*, volume 2567 of *LNCS*, pages 18–30. Springer-Verlag, 2003.

[7] D. Chaum. Blind signatures for untraceable payments. In *Proc. of Crypto'82*, pages 199–203. New York: Plenum Press, 1983.

[8] L. Chen and J. Malone-Lee. Improved identity-based signcryption. In *Public Key Cryptography (PKC 2005)*.

[9] C. Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363. Springer-Verlag, 2001.

[10] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Information Theory*, 31(4):469–472, 1985.

[11] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — Crypto '86*, pages 186–194. Springer-Verlag, 1987.

[12] W. Gao, X. Wang, G. Wang, and F.Li. One-round ID-based blind signature scheme without ROS assumption. Cryptology ePrint Archive, Report. http://eprint.iacr.org/2007/007.

[13] L. C. Guillou and J. J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. In *CRYPTO '88: Proceedings on Advances in cryptology*, pages 216–231. Springer-Verlag, 1990.

[14] F. Hess. Efficient identity based signature schemes based on pairings. In *Proc. of SAC'02*, volume 2595 of *LNCS*, pages 310–324. Springer-Verlag, 2003.

[15] P. Horster, M. Michels, and H. Petersen. Meta-message recovery and meta-blind signature schemes based on the discrete logarithm problem and their applications. In *Proc. of ASIACRYPT 1994*. LNCS, Springer-Verlag, 1994.

[16] P. Horster, M. Michels, and H. Petersen. Meta signature schemes giving message recovery based on the discrete logarithm problem. In *Proc. of 2nd Int. Workshop on IT-Security*, Vienna, 1994.

[17] P. Horster, H. Petersen, and M. Michels. Meta-elgamal signature schemes. In *Proc. of ACM Conference on Computer and Communications Security*, pages 96–107, 1994.

[18] Z. Huang, K. Chen, and Y. Wang. Efficient identity-based signatures and blind signatures. In *Proc. of CANS 2005*, volume 3810 of *LNCS*, pages 120–133, 2005.

[19] A. Joux. A one round protocol for tripartite diffie-hellman. In *Proc. of ANTS-IV*, volume 1838 of *LNCS*, pages 385–394, 2000.

[20] S. Kalkan, K. Kaya, and A. A. Selcuk. Generalized ID-based ElGamal signatures. In *The 22nd International Symposium on Computer and Information Sciences (ISCIS 2007)*, 2007.

[21] B. Libert and J. J. Quisquater. New identity based signcryption schemes from pairings. In *In Information Theory Workshop, 2003*, pages 155–158, 2003.

[22] J. Malone-Lee. Identity based signcryption, 2002. Cryptology ePrint Archive, Report. http://eprint.iacr.org/2002/098.

[23] N. McCullagh and P. S. L. M. Barreto. Efficient and forward-secure identity-based signcryption, 2004. Cryptology ePrint Archive, Report. http://eprint.iacr.org/2004/117.

[24] A. Menezes, S. Vanstone, and T. Okamoto. Reducing elliptic curve logarithms to logarithms in a finite field. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 80–89. ACM, 1991.

[25] A. Miyaji. A message recovery signature scheme equivalent to dsa giving message recovery. In *Proc. of ASIACRYPT'96*, volume 1163 of *LNCS*, pages 1–14, 1996.

[26] D. Nalla and K. Reddy. Signcryption scheme for identity-based cryptosystems, 2003. Cryptology ePrint Archive, Report. http://eprint.iacr.org/2003/066.

[27] K. Nyberg and R. A. Rueppel. A new signature scheme based on the dsa giving message recovery. In *Proc. of 1st ACM conference on communication and computer security*, pages 58–61, 1993.

[28] K. Nyberg and R. A. Rueppel. Message recovery for signature schemes based on the discrete logarithm problem. In *Proc. of EUROCRYPT'94*, volume 950 of *LNCS*, pages 182–193, 1995.

[29] K. Paterson. Id-based signatures from pairings on elliptic curves, 2002. Cryptology ePrint Archive, Report. http://eprint.iacr.org/2002/004.

[30] K. G. Paterson and J. C. N. Schuldt. Efficient identity-based signatures secure in the standard model. In *In L.M. Batten and R. Safavi-Naini (eds.), ACISP 2006*, volume 4058 of *LNCS*, pages 207–222. Springer-Verlag, 2006.

[31] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[32] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Proc. of SCIS'00*, 2003.

[33] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer-Verlag, 1984.

[34] J.-B. Shin, K. Lee, and K. Shim. New DSA-Verifiable signcryption schemes. In *Proc. of Information Security and Cryptology - ICISC 2002*, volume 2587 of *LNCS*, pages 35–47. Springer-Verlag, 2002.

[35] R. Steinfeld and Y. Zheng. A signcryption scheme based on integer factorization. In *ISW '00: Proceedings of the Third International Workshop on Information Security*, volume 1975, pages 308–322. Springer-Verlag, 2000.

[36] R. Tso, C. Gu, T. Okamoto, and E. Okamoto. An efficient ID-based digital signature with message recovery based on pairing. http://citeseer.ist.psu.edu/tso06efficient.html.

[37] C. Y. Yeun. Digital signature with message recovery and authenticated encryption (signcryption)– a comparison. In *IMA - Cryptography and Coding'99*, volume 1746 of *LNCS*, pages 307–312, 1999.

[38] X. Yi. An identity based signature scheme from the weil pairing. *IEEE Communication Letters*, 7(2):76–78, 2003.

[39] D. H. Yum and P. J. Lee. New signcryption schemes based on kcdsa. In *ICISC '01: Proceedings of the 4th International Conference Seoul on Information Security and Cryptology*, volume 2288, pages 305–317. Springer-Verlag, 2002.

[40] F. Zhang and K. Kim. ID-based blind signature and ring signature from pairings. In *Proc. of Asiacript 2002*, volume 2501 of *LNCS*, pages 533–547, 2002.

[41] F. Zhang and K. Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. In *Proc. of ACISP2003*, volume 2727 of *LNCS*, pages 312–323, 2003.

[42] F. Zhang, W. Susilo, and Y. Mu. Identity-based partial message recovery signatures. In *Financial Cryptography'05*, volume 3570 of *LNCS*, pages 45–56, 2005.

[43] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, volume 1294, pages 165–179. Springer-Verlag, 1997.

[44] Y. Zheng. Signcryption and its applications in efficient public key solutions. In *ISW '97: Proceedings of the First International Workshop on Information Security*, volume 1396, pages 291–312. Springer-Verlag, 1998.

[45] Y. Zheng. Identification, signature and signcryption using high order residues modulo an RSA composite. In *PKC '01: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography*, volume 1992, pages 48–63. Springer-Verlag, 2001.

[46] Y. Zheng and H. Imai. Efficient signcryption schemes on elliptic curves. In *Proc. of IFIP SEC'98*, 1998.