An Efficient Signature Scheme from Bilinear Pairings and Its Applications

Fangguo Zhang, Reihaneh Safavi-Naini and Willy Susilo

School of Information Technology and Computer Science
University of Wollongong, NSW 2522 Australia
{fangguo, rei, wsusilo}@uow.edu.au

Abstract. In Asiacrypt2001, Boneh, Lynn, and Shacham [8] proposed a short signature scheme (BLS scheme) using bilinear pairing on certain elliptic and hyperelliptic curves. Subsequently numerous cryptographic schemes based on BLS signature scheme were proposed. BLS short signature needs a special hash function [6, 1, 8]. This hash function is probabilistic and generally inefficient. In this paper, we propose a new short signature scheme from the bilinear pairings that unlike BLS, uses general cryptographic hash functions such as SHA-1 or MD5, and does not require special hash functions. Furthermore, the scheme requires less pairing operations than BLS scheme and so is more efficient than BLS scheme. We use this signature scheme to construct a ring signature scheme and a new method for delegation. We give the exact security proofs for the new signature scheme and the ring signature scheme in the random oracle model.

Keywords: *Short signature, Bilinear pairings, ID-based cryptography, Ring signature, Proxy signature*

1 Introduction

In recent years, bilinear pairings have found various applications in cryptography and have allowed us to construct some new cryptographic schemes [5–8, 11, 20, 23, 27]. In these schemes, there is a basic signature scheme, the BLS scheme, that has the shortest length among signature schemes in classical cryptography. The scheme is based on Weil pairing and can be obtained from the private key extraction process of Boneh-Franklin's [6] ID-based encryption scheme. BLS short signature needs a special hash function, i.e., an admissible encoding function called MapToPoint that is also used by most conventional cryptographic schemes from pairings. Although there has been much discussion on the construction of such hash algorithm [1, 8], these algorithms are still probabilistic, and to our knowledge there is no deterministic polynomial time algorithm for them.

The Computational Diffie-Hellman Problem (CDHP) is a well-studied problem and its hardness is widely believed to be closely related to the hardness of the

Discrete Logarithm Problem (DLP). There are two variations of CDHP: Inverse Computational Diffie-Hellman Problem (Inv-CDHP) and Square Computational Diffie-Hellman Problem (Squ-CDHP).

In this paper, we propose a new short signature scheme from the bilinear pairings. Our scheme is constructed from Inv-CDHP based on bilinear pairing and does not require any special hash function. We note that the computation of the pairing is the most time-consuming in pairing based cryptosystems. Although there have been many papers discussing the complexity of pairings and how to speed up the pairing computation [2, 11], the computation of the pairing still remains time-consuming. Our new scheme uses less pairing operations than BLS scheme, and hence, is more efficient than BLS scheme. Based on the new signature scheme, we propose a ring signature scheme and a new method for delegation (so far some proxy signature schemes). We give exact security proofs for the new signature scheme and the ring signature scheme in the random oracle model (the cryptographic hashing function (such as MD5 or SHA-1) is seen as an oracle which produces a random value for each new query).

The rest of the paper is organized as follows: The next section briefly explains the bilinear pairing and some problems related to pairings. Section 3 gives the new basic signature scheme and its security analysis. Section 4 briefly shows the relationship between the ID-based public key setting and this basic signature scheme. Based on the basic signature scheme, we give a ring signature scheme and some proxy signature schemes in Section 5 and 6, respectively. Section 7 concludes this paper.

2 Bilinear Pairing and Some Problems

Let $\mathbb{G}_1$ be a cyclic additive group generated by P, whose order is a prime q, and $\mathbb{G}_2$ be a cyclic multiplicative group with the same order q. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a map with the following properties:

1. **Bilinearity:** $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1, a, b \in \mathbb{Z}_q$
2. **Non-degeneracy:** There exists $P, Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$, in other words, the map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$;
3. **Computability:** There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

In our setting of prime order groups, the **Non-degeneracy** is equivalent to $e(P, Q) \neq 1$ for all $P, Q \in \mathbb{G}_1$. So, when P is a generator of $\mathbb{G}_1$, $e(P, P)$ is a generator of $\mathbb{G}_2$. Such a bilinear map is called a bilinear pairing (more exactly, called an admissible bilinear pairing).

We consider the following problems in the additive group $(\mathbb{G}_1; +)$.

- **Discrete Logarithm Problem (DLP):** Given two group elements P and Q, find an integer $n \in \mathbb{Z}_q^*$, such that $Q = nP$ whenever such an integer exists.
- **Decision Diffie-Hellman Problem (DDHP):** For $a, b, c \in \mathbb{Z}_q^*$, given P, aP, bP, cP decide whether $c \equiv ab \bmod q$.

- **Computational Diffie-Hellman Problem (CDHP):** For $a, b \in \mathbb{Z}_q^*$, given P, aP, bP, compute abP.

There are two variations of CDHP:

- **Inverse Computational Diffie-Hellman Problem (Inv-CDHP):** For $a \in \mathbb{Z}_q^*$, given P, aP, to compute $a^{-1}P$.
- **Square Computational Diffie-Hellman Problem (Squ-CDHP):** For $a \in \mathbb{Z}_q^*$, given P, aP, to compute a^2P.

The following theorem relates these problemes [17, 22].

Theorem 1. *CDHP, Inv-CDHP and Squ-CDHP are polynomial time equivalent.*

From the bilinear pairing, we have another variant of the CDHP called the bilinear Diffie-Hellman problem (BDHP).

Definition 1. *The BDHP in* $(\mathbb{G}_1, \mathbb{G}_2, e)$ *is defined as follows: given* (P, aP, bP, cP) *for some* $a, b, c \in \mathbb{Z}_q^*$*, compute* $v \in \mathbb{G}_2$ *such that* $v = e(P, P)^{abc}$*.*

Most previous cryptographic schemes from bilinear pairings depend on BDHP, such as Joux's one round tripartite Diffie-Hellman protocol [14] and Boneh-Franklin's [6] ID-based encryption scheme. It is not hard to obtain two variations of BDHP:

- **Bilinear Inverse Diffie-Hellman Problem (BIDHP):** For $a, c \in \mathbb{Z}_q^*$, given P, aP, cP, to compute $v = e(P, P)^{a^{-1}c}$.
- **Bilinear Square Diffie-Hellman Problem (BSDHP):** For $a \in \mathbb{Z}_q^*$, given P, aP, cP, to compute $v = e(P, P)^{a^2c}$.

Like Theorem 1, we have the following theorem. A simple proof is given in Appendix A.

Theorem 2. *BDHP, BSDHP and BIDHP are polynomial time equivalent.*

Assumptions: We assume that DLP, CDHP Inv-CDHP, Squ-CDHP, BDHP, BIDHP and BSDHP are hard, which means there is no polynomial time algorithm to solve any of them with non-negligible probability.

A *Gap Diffie-Hellman* (*GDH*) *group* is a group which the DDHP is easy but the CDHP is hard in it. From bilinear pairing, we can obtain the GDH group. Such groups can be found on supersingular elliptic curves or hyperelliptic curves over finite field, and the bilinear parings can be derived from the Weil or Tate pairing. For more details, we refer the readers to [6, 9, 13].

All schemes in this paper can work on any GDH group. Throughout this paper, we define the system parameters in all schemes as follows. Let P be a generator of $\mathbb{G}_1$ with order q, the bilinear pairing is given by $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. These system parameters can be obtained using a **GDH Parameter Generator** $\mathcal{IG}$ [6]. Define a cryptographic hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^\lambda$, where $|q| \geq \lambda \geq 160$.

3 New Short Signature Scheme from Bilinear Pairings

3.1 The Basic Signature Scheme

A signature scheme consists of the following four algorithms : a parameter generation algorithm $\mathsf{ParamGen}$, a key generation algorithm KeyGen, a signature generation algorithm Sign and a signature verification algorithm Ver.

We describe the new signature scheme as follows:

1. $\mathsf{ParamGen}$. The system parameters are $\{\mathbb{G}_1, \mathbb{G}_2, e, q, P, H\}$.
2. KeyGen. Randomly selects $x \in_R \mathbb{Z}_q^*$, and computes $P_{pub} = xP$. The public key is P_{pub}. The secret key is x.
3. Sign. Given a secret key x, and a message m, computes $S = \frac{1}{H(m)+x}P$. The signature is S.
4. Ver. Given a public key P_{pub}, a message m, and a signature S, verify if

$$e(H(m)P + P_{pub},\ S) = e(P,\ P).$$

The verification works because of the following equations:

$$\begin{aligned} e(H(m)P + P_{pub},\ S) &= e((H(m)+x)P,\ (H(m)+x)^{-1}P) \\ &= e(P,\ P)^{(H(m)+x)\cdot(H(m)+x)^{-1}} \\ &= e(P,\ P) \end{aligned}$$

3.2 Security Discussions

The strongest notion of security for signature schemes was defined by Goldwasser, Micali and Rivest [12] as follows:

Definition 2 (Secure signatures [12]). *A signature scheme $\mathcal{S} = <$ $\mathsf{ParamGen}$, KeyGen, Sign, Ver $>$ is existentially unforgeable under an adaptive chosen message attack if it is infeasible for a forger who only knows the public key to produce a valid message-signature pair after obtaining polynomially many signatures on messages of its choice from the signer.*

Formally, for every probabilistic polynomial time forger algorithm $\mathcal{F}$ there does not exist a non-negligible probability ϵ such that

$$Adv(\mathcal{F}) = \Pr \begin{bmatrix} \langle pk, sk \rangle \leftarrow \langle \mathsf{ParamGen}, \mathsf{KeyGen} \rangle (1^l); \\ for i = 1, 2, \ldots, k; \\ m_i \leftarrow \mathcal{F}(pk, m_1, \sigma_1, \ldots, m_{i-1}, \sigma_{i-1}), \sigma_i \leftarrow \mathsf{Sign}(sk, m_i); \\ \langle m, \sigma \rangle \leftarrow \mathcal{F}(pk, m_1, \sigma_1, \ldots, m_k, \sigma_k); \\ m \notin \{m_1, \ldots, m_k\} \text{ and } \mathsf{Ver}(pk, m, \sigma) = accept \end{bmatrix} \geq \epsilon.$$

Here we use the definition of [4] which takes into account the presence of an ideal hash function, and gives a concrete security analysis of digital signatures.

Definition 3 (Exact security of signatures [4]). *A forger $\mathcal{F}$ is said to (t, q_H, q_S, ϵ)-break the signature scheme $\mathcal{S} = <$ ParamGen, KeyGen, Sign, Ver $>$ via an adaptive chosen message attack if after at most q_H queries to the hash oracle, q_S signatures queries and t processing time, it outputs a valid forgery with probability at least ϵ.*

A signature scheme $\mathcal{S}$ is (t, q_H, q_S, ϵ)-secure if there is no forger who (t, q_H, q_S, ϵ)-breaks the scheme.

To give the security proof of the new signature scheme, we introduce a problem proposed by S. Mitsunari *et. al* [18]. The problem was called k-CAA (collusion attack algorithm with k traitors) in Mitsunari *et. al*'s traitor tracing scheme.

Definition 4 (k-CAA). *For an integer k, and $x \in_R \mathbb{Z}_q$, $P \in \mathbb{G}_1$, given*

$$\{P, Q = xP, h_1, \ldots, h_k \in \mathbb{Z}_q, \frac{1}{h_1 + x}P, \ldots, \frac{1}{h_k + x}P\},$$

to compute $\frac{1}{h+x}P$ for some $h \notin \{h_1, \ldots, h_k\}$.

We say that the k-CAA is (t, ϵ)-hard if for all t-time adversaries $\mathcal{A}$, we have

$$Adv\text{k-CAA}_{\mathcal{A}} = \Pr\begin{bmatrix} \mathcal{A}(P, Q = xP, \frac{1}{h_1+x}P, \ldots, \frac{1}{h_k+x}P) = \frac{1}{h+x}P \\ |x \in_R \mathbb{Z}_q, P \in \mathbb{G}_1, h_1, \ldots, h_k \in \mathbb{Z}_q, h \notin \{h_1, \ldots, h_k\} \end{bmatrix} < \epsilon.$$

About the security of proposed signature scheme against an adaptive chosen message attack, we have the following theorem:

Theorem 3. *If there exists a (t, q_H, q_S, ϵ)-forger $\mathcal{F}$ using adaptive chosen message attack for the proposed signature scheme, then there exists a (t', ϵ')-algorithm $\mathcal{A}$ solving q_S-CAA, where $t' = t$, $\epsilon' \geq (\frac{q_S}{q_H})^{q_S} \cdot \epsilon$* [1]*.*

Proof. In the proposed signature scheme, before signing a message m, we need to make a query $H(m)$. Our proof is in random oracle model (the hash function is seen as a random oracle, i.e., the output of the hash function is uniformly distributed).

Suppose that a forger $\mathcal{F}$ (t, q_H, q_S, ϵ)-break the signature scheme using an adaptive chosen message attack. We will use $\mathcal{F}$ to construct an algorithm $\mathcal{A}$ to solve q_S-CAA.

Suppose $\mathcal{A}$ is given a challenge:

" Given $P \in \mathbb{G}_1$, $Q = xP$, $h_1, h_2, \ldots, h_{q_S} \in \mathbb{Z}_q$, and $\frac{1}{h_1+x}P$, $\frac{1}{h_2+x}P, \ldots, \frac{1}{h_{q_S}+x}P$, to compute $\frac{1}{h+x}P$ for some $h \notin \{h_1, \ldots, h_{q_S}\}$."

Now $\mathcal{A}$ plays the role of the signer and sets $P_{pub} = Q$. $\mathcal{A}$ will answer hash oracle queries and signing queries itself. We assume that $\mathcal{F}$ never repeats a hash query or a signature query.

S1 $\mathcal{A}$ prepares q_H responses $\{w_1, w_2, \ldots, w_{q_H}\}$ of the hash oracle queries, $h_1, \ldots, h_{q_S}$ are distributed randomly in this response set.

[1] To obtain a good bound for ϵ', we should assume that q_S and q_H are very closed.

S2 $\mathcal{F}$ makes a hash oracle query on m_i for $1 \leq i \leq q_H$. $\mathcal{A}$ sends w_i to $\mathcal{F}$ as the response of the hash oracle query on m_i.

S3 $\mathcal{F}$ makes a signature oracle query for w_i. If $w_i = h_j$, $\mathcal{A}$ returns $\frac{1}{h_j+x}P$ to $\mathcal{F}$ as the response. Otherwise the process stops and $\mathcal{A}$ has failed.

S4 Finally $\mathcal{F}$ halts and outputs a message-signature pair (m, S). Here the hash value of m is some w_l and $w_l \notin \{h_1, \ldots, h_{q_S}\}$. Since (m, S) is a valid forgery and $H(m) = w_l$, it satisfies:

$$e(H(m)P + Q,\ S) = e(P,\ P).$$

So, $S = \frac{1}{w_l+x}P$. $\mathcal{A}$ outputs (w_l, S) as a solution to $\mathcal{A}$'s challenge.

Algorithm $\mathcal{F}$ cannot distinguish between $\mathcal{A}$'s simulation and real life because the hash function behaves as a random oracle. The running time of $\mathcal{A}$ is equal to the running time of $\mathcal{F}$ $t' = t$. In step S3, the success probability of $\mathcal{A}$ is $\frac{q_S}{q_H}$, so, for all signature oracle queries, $\mathcal{A}$ will not fail with probability $\rho \geq (\frac{q_S}{q_H})^{q_S}$ (if $\mathcal{F}$ only makes $s(\leq q_S)$ signature oracle queries, the success probability of $\mathcal{A}$ is $(\frac{q_S}{q_H})^s$). Hence, after the algorithm $\mathcal{A}$ finished step S4, the success probability of $\mathcal{A}$ is:

$$\epsilon' \geq (\frac{q_S}{q_H})^{q_S} \cdot \epsilon.$$

□

In [18], S. Mitsunari *et. al* introduced another new problem, k-weak Computational Diffie-Hellman Problem (k-wCDHP), and gave the following theorem.

Definition 5 (k-wCDHP). *Given $k+1$ values $< P, yP, y^2P, \ldots, y^kP >$, to compute $\frac{1}{y}P$.*

Theorem 4 ([18]). *There exists a polynomial time algorithm to solve (k-1)-wCDHP if and only if there exists a polynomial time algorithm for k-CAA.*

So, in our signature scheme, the security against the existential forgery under an adaptive chosen message attack at least depends on k-wCDHP.

To give a more specific evaluation of the security of our signature scheme, we introduce a new problem.

Definition 6 (k+1 Exponent Problem). *Given $k+1$ values $< P, yP, y^2P, \ldots, y^kP >$, to compute $y^{k+1}P$.*

We have the following theorem.

Theorem 5. *k-wCDHP and k+1EP are polynomial time equivalent.*

Proof. k-wCDHP $\Rightarrow k+1EP$:

Given $k+1$ values $P,\ yP,\ y^2P,\ \ldots,\ y^kP$, let $Q = y^kP,\ tQ = y^{k-1}P$, and so $t = y^{-1}$.

Set the input of k-wCDHP to be

$$Q = y^kP, y^{k-1}P = tQ, y^{k-2}P = t^2Q, \ldots, yP = t^{k-1}Q, t^kQ.$$

Then k-wCDHP outputs

$$t^{-1}Q = (y^{-1})^{-1}y^kP = y^{k+1}P.$$

$k+1EP \Rightarrow$ k-wCDHP :

Given $k+1$ values $P,\ yP,\ y^2P,\ \ldots,\ y^kP$, let $Q = y^kP,\ tQ = y^{k-1}P$, and so $t = y^{-1}$.

Set the input of $k+1EP$ to be

$$Q = y^kP, y^{k-1}P = tQ, y^{k-2}P = t^2Q, \ldots, yP = t^{k-1}Q, t^kQ.$$

Then $k+1EP$ outputs

$$t^{k+1}Q = y^{-k-1}y^kP = y^{-1}P.$$

□

We note that $k+1EP$ and k-wCDHP are no harder than the CDHP. There exists a special case where k-wCDHP or $k+1EP$ can be easily solved: Given

$$P_0 = P, P_1 = yP, P_2 = y^2P, \ldots, P_k = y^kP,$$

if there are at least two same elements in them, *e.g.*, $P_i = P_j\ (i \neq j)$, that means $y^i \bmod q \equiv y^j \bmod q$, and so, the order of y in $\mathbb{Z}_q$ is $j-i$. Then

$$y^{-1}P = P_{j-i-1} \quad or \quad y^{k+1}P = P_{k+1 \bmod (j-i)}.$$

This case gives an attack on the new signature scheme. However, because y can be regarded as a random element in $\mathbb{Z}_q^*$, we can show that the success probability of this attack is negligible.[2]

Let $q-1 = \prod_{i=1}^{s} p_i^{e_i}$. For any $a \in \mathbb{Z}_q^*$, the order of a is a divisor of $q-1$. Given k, suppose that the number of element a in $\mathbb{Z}_q^*$ such that $ord(a) \leq k$ is given by N. Obviously, $N < k^2$ (the maximum of the number of the divisors less than k is k). Let ρ be the probability that a randomly chosen element in $\mathbb{Z}_q^*$ has order less than k, then

$$\rho = \frac{N}{q} < \frac{k^2}{q}.$$

So, if $q \approx 2^{160}$, we limit $k \leq 2^{40}$, which means the attacker has at most 2^{40} message-signature pairs. Then using the above attack, the success probability is at most

$$\frac{(2^{40})^2}{2^{160}} = 2^{-80} \approx 0.82718 \times 10^{-24}.$$

Summarizing the above discussions, we have the following result.

Corollary 1 *Assuming that k+1EP is hard, i.e., there is no polynomial time algorithm to solve k+1EP with non-negligible probability, then the proposed signature scheme is secure under the random oracle model.*

[2] If q is Sophie-German prime, then it is obvious that this attack is impossible. However, in practice, $\mathbb{G}_1$ may be an elliptic curve group or hyperelliptic curve Jacobian, it is hard to find $\mathbb{G}_1$ such that the order of $\mathbb{G}_1$ is a Sophie-German prime. So, we only assume that the order of $\mathbb{G}_1$ is prime, not any special prime.

3.3 Efficiency

Short signatures are important in low-bandwidth communication environments. A number of short signature schemes, such as: Quartz [19], McEliece-based signature [10], have been proposed. BLS scheme is the shortest signature scheme known in classical cryptography (Quartz and McEliece-based signature belong to the multivariate cryptography). Our signature only consists of one element of $\mathbb{G}_1$. In practice, the size of the element in $\mathbb{G}_1$ (elliptic curve group or hyperelliptic curve Jacobians) can be reduced by a factor of 2 using compression techniques. So, like BLS signature scheme, our signature scheme is a short signature scheme.

We compare our signature scheme with the BLS scheme from computation overhead view point. We denote `Pa` the pairing operation, `Pm` the point scalar multiplication on $\mathbb{G}_1$, `Ad` the point addition on $\mathbb{G}_1$, `Inv` the inversion in $\mathbb{Z}_q$ and `MTP` the MapToPoint hash operation in BLS scheme. We summarize the result in Table 1(we ignore the general hash operation).

Schemes	*Setup*	*Signing*	*Verification*
Proposed	*Same*	$1\mathtt{Inv} + 1\mathtt{Pm}$	$2(or\ 1)\mathtt{Pa} + 1\mathtt{Pm} + 1\mathtt{Ad}$
BLS scheme	*Same*	$1\mathtt{MTP} + 1\mathtt{Pm}$	$2\mathtt{Pa} + 1\mathtt{MTP}$

Table 1. Comparison of our scheme and the BLS scheme

We assume that BLS scheme and our scheme are all using the GDH group derived from the curve $E/F_{3^{163}}$ defined by the equation $y^2 = x^3 - x + 1$. The group provides 1551-bit discrete-log security. The MapToPoint hash operation requires at least one quadratic or cubic equation over $F_{3^{163}}$ to be solved. So the cost of one MapToPoint hash operation is bigger than one inversion in $\mathbb{Z}_q$. Despite a number of attempts [2, 3, 11] to reduce the complexity of pairing, still the operation is very costly. For example, according to the best result in [3], one pairing operation is about 11110 multiplications in $F_{3^{163}}$, while a point scalar multiplication of $E/F_{3^{163}}$ is a few hundred multiplications in $F_{3^{163}}$. In our scheme, $e(P, P)$ can be precomputed and published as part of the signer's public key and so there is only one pairing operation in verification. This compare to two pairing operations in BLS scheme, gives a more efficient scheme.

4 Relation to ID-based Public Key Setting

The concept of ID-based encryption and signature were first introduced by Shamir [26]. The basic idea of ID-based cryptosystems is to ues the identity information of a user functions as his public key. ID-based public key setting involves a Private Key Generator (PKG) and users. The basic operations consist of **setup** and **private key extraction**. Informally, an ID-based encryption scheme (IBE) consists of four algorithms: (1) **Setup** generates the system parameters and a *master-key*, (2) **Extract** uses the *master-key* to generate the private key corresponding to an arbitrary string ID, (3) **Encrypt** encrypts a plaintext using a public key ID and (4) **Decrypt** decrypts the ciphertexts using the corresponding private key.

Recently, bilinear pairings have been used to construct ID-based cryptosystem. As noted by Moni Naor in [6], any ID-based encryption scheme immediately gives a public key signature scheme. Therefore, there is a relationship between the short signature schemes and the ID-based public key setting from bilinear pairing, that is the signing process in the short signature scheme can be regarded as the private key extract process in the ID-based public key setting. From this viewpoint, our new signature scheme can be regarded as being derived from Sakai-Kasahara's new ID-based encryption scheme with pairing [24, 25].

5 A Ring Signature Scheme

Ring signature schemes were proposed by Rivest, Shamir, and Tauman [21]. In a ring signature, a user selects a set of possible signers including himself that is called a ring. A possible signer is anyone with a public key for a standard signature scheme. The user can then sign a message using his private key and the public keys of all of the members of the ring. The signed message then has the property that it can be verified to be signed by a user in the ring, but the identity of the actual signer will not be revealed, hence the signature provides anonymity for the signer and the anonymity cannot be revoked.

Ring signature schemes should satisfy the following properties: **Correctness**, **Unconditional ambiguity** or **Anonymity** and **Unforgeability**.

A number of ring signature schemes based on the pairings are proposed. Zhang et.al [28] proposed an ID-based ring signature scheme. In [7], Boneh et.al gave a ring signature scheme from BLS signature scheme. In this section, we give a new ring signature scheme based on the signature scheme in Section 3.

The system parameters are $params = \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, H\}$. Let Alice be a signer with public key $P_{pubk} = s_k P$ and private key s_k, and $L = \{P_{pubi}\}$ be the set of public keys and $|L| = n$.

Ring Signing:

For message m, Alice chooses $a_i \in_R \mathbb{Z}_q$ for all $i \neq k$ and obtains

$$S_k = -\frac{1}{H(m) + s_k} \sum_{i \neq k} (a_i(H(m)P + P_{pubi})) + \frac{1}{H(m) + s_k} P.$$

Let $S_i = a_i P$, for all $i \neq k$. The ring signature is $\sigma = \langle S_1, S_2, \ldots, S_n \rangle$.

Ring Verification:

$$\prod_{i=1}^{n} e(H(m)P + P_{pubi},\ S_i) = e(P,\ P).$$

The following is a brief analysis of the scheme.

Correctness. The verification of the signature is correct because of the following.

$$\begin{aligned}
&\prod_{i=1}^{n} e(H(m)P + P_{pubi},\ S_i)\\
&= \prod_{i \neq k} e(H(m)P + P_{pubi},\ S_i) \cdot e(H(m)P + P_{pubk},\ S_k)\\
&= \prod_{i \neq k} e(H(m)P + P_{pubi},\ a_iP) \cdot e(H(m)P + P_{pubk},\ \frac{1}{H(m) + s_k}(P -\\
&\quad \sum_{i \neq k}(a_i(H(m)P + P_{pubi})))\\
&= e(\sum_{i \neq k}(a_i(H(m)P + P_{pubi})),\ P) \cdot e(P,\ -\sum_{i \neq k}(a_i(H(m)P + P_{pubi}))) \cdot e(P,\ P)\\
&= e(P,\ P)
\end{aligned}$$

Unconditional ambiguity. The scheme has unconditionally signer-ambiguity. Assume that $\sigma = \langle S_1, S_2, \ldots, S_n \rangle$ is a ring signature on the set of users L generated with private key s_k. All S_i except S_k are taken randomly from $\mathbb{G}_1$ due to $S_i = a_iP$ and $a_i \in_R \mathbb{Z}_q$. S_k is computed by these $a_i, H(m)$ and s_k. Therefore, for fixed L and m, $\langle S_1, S_2, \ldots, S_n \rangle$ has $\mid \mathbb{G}_1 \mid^{n-1}$ possible values, all of which can be chosen by the signature generation procedure with equal probability and regardless of the signer. At the same time, the distribution $\{S_1, S_2, \ldots, S_n\}$ is identical to the distribution $\{a_1P, a_2P, \ldots, a_nP : \sum_{i=1}^{n} a_iP = C\}$, here C is element of $\mathbb{G}_1$ depend on L and m. So, for any algorithm $\mathcal{A}$, any set of users L, and a random $k \in L$, the probability $\Pr[\mathcal{A}(\sigma) = k]$ is at most $1/ \mid L \mid$.

Unforgeability. For the unforgeability, we have the following theorem:

Theorem 6. *If there exists a (t, q_H, q_S, ϵ)-forger $\mathcal{F}$ algorithm that can produce a forgery of a ring signature on a set of users of size n, then there exists a (t', ϵ')-algorithm $\mathcal{A}$ that can solve q_S-CAA, where*

$$t' \leq t + (3 + q_S)nt_{sm} + 2(n-1)t_{add} + (n-1)t_{mu} + (n-1)t_{inv},$$

$$\epsilon' \geq (\frac{q_S}{q_H})^{q_S} \cdot \frac{1}{q_H - q_S} \cdot \epsilon.$$

Here, t_{sm} is the time of one point scalar multiplication in $\mathbb{G}_1$, t_{add} is the time of one addition in $\mathbb{G}_1$, t_{inv} is the time of one inversion in $\mathbb{Z}_q$ and t_{mu} is the time of one multiplications in $\mathbb{Z}_q$.

Proof. We adopt the security model of Rivest, Shamir and Tauman. Consider the following game played between an adversary and a challenger. The adversary is given the public keys $P_1, \ldots, P_n$ of a set of users U , and is given oracle access to H and a ring-signing oracle. The goal of the adversary is to output a valid ring signature on U of a message m subject to the condition that m has never been presented to the ring-signing oracle.

Suppose that there exists a (t, q_H, q_S, ϵ)-forger $\mathcal{F}$ algorithm that can produce a forgery of a ring signature on a set of users of size n. We will use $\mathcal{F}$ to construct an algorithm $\mathcal{A}$ to solve q_S-CAA. Suppose that $\mathcal{A}$ is given a challenge:

"Given $P \in \mathbb{G}_1, Q = xP, h_1, h_2, \ldots, h_{q_S} \in \mathbb{Z}_q$, and $\frac{1}{h_1+x}P, \frac{1}{h_2+x}P, \ldots, \frac{1}{h_{q_S}+x}P$, compute $\frac{1}{h+x}P$ for some $h \notin \{h_1, \ldots, h_{q_S}\}$"

– **Setup:** $\mathcal{A}$ plays the role of the real signer and picks $a_1 = 1, a_2, \ldots, a_n$ at random from $\mathbb{Z}_q$ and sets

$$P_1 = Q,\ P_2 = a_2Q + h(a_2 - 1)P,\ \ldots,\ P_n = a_nQ + h(a_n - 1)P.$$

Here, we assume that the number of users n is an odd number. $\mathcal{A}$ prepares q_H respondences $\{w_1, w_2, \ldots, w_{q_H}\}$ of hash oracle queries. $h_1, \ldots, h_{q_S}$ and h are distributed randomly in this respondences set.

– **Hash queries:** $\mathcal{F}$ is given the public keys $P_1, P_2, \ldots, P_n$. $\mathcal{F}$ makes a hash oracle query on m_i for $1 \leq i \leq q_H$. $\mathcal{A}$ sends w_i to $\mathcal{F}$ as the respondence of hash oracle query on m_i.

– **Signing queries:** $\mathcal{F}$ makes a ring signature oracle query for w_i. If $w_i = h_j$, $\mathcal{A}$ returns

$$\sigma_i = \{S_{i1},\ S_{i2},\ \ldots,\ S_{in}\}$$

to $\mathcal{F}$ as the signing result. Here

$$\begin{aligned}
S_{i1} &= (1 - \sum_{l=2}^{n}(-1)^l(a_l - 1)^{-1}) \cdot \frac{1}{h_j + x}P = (1 - a) \cdot \frac{1}{h_j + x}P \\
S_{i2} &= (a_2 - 1)^{-1} \cdot \frac{1}{h_j + x}P \\
S_{i3} &= -(a_3 - 1)^{-1} \cdot \frac{1}{h_j + x}P \\
\ldots &= \ \ldots \\
S_{il} &= (-1)^l(a_l - 1)^{-1} \cdot \frac{1}{h_j + x}P \\
\ldots &= \ \ldots \\
S_{in} &= (a_n - 1)^{-1} \cdot \frac{1}{h_j + x}P
\end{aligned}$$

From the construction of S_{il}, we can verify that σ_i can pass the ring verification:

$$\begin{aligned}
&\prod_{l=1}^{n} e(H(m_i)P + P_l, S_{il}) \\
&= e(h_jP + Q, \frac{1-a}{h_j + x}P) \prod_{l=2}^{n} e(h_jP + a_lQ + h(a_l - 1)P, \frac{(-1)^t(a_l - 1)^{-1}}{h_j + x}P)
\end{aligned}$$

$$= e(P,P)^{1-a}\prod_{l=2}^{n} e(h_jP + Q + (a_l-1)Q + h(a_l-1)P, \frac{(-1)^t(a_l-1)^{-1}}{h_j+x}P)$$
$$= e(P,P)^{1-a}\prod_{l=2}^{n} e(h_jP + Q, \frac{(-1)^t(a_l-1)^{-1}}{h_j+x}P)e((a_l-1)(Q+hP),$$
$$\frac{(-1)^t(a_l-1)^{-1}}{h_j+x}P)$$
$$= e(P,P)^{1-a}\prod_{l=2}^{n} e(P,P)^{(-1)^t(a_l-1)^{-1}} \ (\textit{Due to } n \textit{ be an odd number})$$
$$= e(P,P)$$

Otherwise, the process stops and $\mathcal{A}$ reports failure.

- **Output:** Eventually $\mathcal{F}$ outputs a message-signature pair $(m, \sigma = \{S_1, S_2, \ldots, S_n\})$ for ring public keys $P_1, P_2, \ldots, P_n$, here the hash value of m is some w_l such that no signature query was issued for m. If $w_l \neq h$, then $\mathcal{A}$ reports failure and terminates. Otherwise,

$$\prod_{i=1}^{n} e(H(m)P + P_i, S_i) = \prod_{i=1}^{n} e(hP + a_iQ + h(a_i-1)P, S_i) = e(P,P).$$

Hence

$$\prod_{i=1}^{n} e(a_ihP + a_iQ, S_i) = \prod_{i=1}^{n} e(hP+Q, a_iS_i) = e(hP+Q, \sum_{i=1}^{n} a_iS_i) = e(P,P).$$

Then $\mathcal{A}$ outputs the required $\frac{1}{h+x}P$ as $\sum_{i=1}^{n} a_iS_i$.

$\mathcal{A}$ will not fail with probability $(\frac{q_S}{q_H})^{q_S} \cdot \frac{1}{q_H - q_S} \cdot \epsilon$ (For all signature oracle queries, $\mathcal{A}$ will not fail with probability $\rho \geq (\frac{q_S}{q_H})^{q_S}$. In **Output**, the probability of $w_l = h$ is $\frac{1}{q_H - q_S}$).

In **Setup**, there are $n-1$ multiplications in $\mathbb{Z}_q$, $n-1$ additions and $2n$ scalar multiplications of $\mathbb{G}_1$. There are nq_S scalar multiplications of $\mathbb{G}_1$ and $n-1$ inversions over $\mathbb{Z}_q$ in $\mathcal{A}$'s signature queries, and n scalar multiplications $n-1$ additions of $\mathbb{G}_1$ in **Output**. We denote t_{sm} the time of one scalar multiplication in $\mathbb{G}_1$, t_{add} the time of one addition in $\mathbb{G}_1$, t_{inv} the time of one inversion in $\mathbb{Z}_q$ and t_{mu} the time of one multiplications in $\mathbb{Z}_q$. So $\mathcal{A}$'s running time t' is $\mathcal{F}$'s running time plus $(2n + nq_S + n)t_{sm} + 2(n-1)t_{add} + (n-1)t_{mu} + (n-1)t_{inv}$, i.e.,

$$t' \leq t + (3 + q_S)nt_{sm} + 2(n-1)t_{add} + (n-1)t_{mu} + (n-1)t_{inv}.$$

□

Note that when $n = 1$, this ring signature scheme is the basic signature scheme.

6 Delegation of Right and Proxy Signatures

Assume that there are two participants, one called original signer with public key PK_o and secret key s_o, another called proxy signer with public key PK_p and secret key s_p, they have the common system parameters: $\{\mathbb{G}_1, \mathbb{G}_2, e, q, P, H\}$. We describe the delegation in detail as follows:

- The original signer makes a warrant w. There is an explicit description of the delegation relation in the warrant w.
- The original signer computes $So_w = (s_o + H(w))^{-1} PK_p$, and sends w and So_w to proxy signer.
- The proxy signer checks if $e(H(w)P + PK_o, So_w) = e(P, PK_p)$, if it is right, then computes $S_w = s_p So_w$.

S_w satisfies: $e(H(w)P + PK_o, S_w) = e(PK_p, PK_p)$.

Anyone cannot forge an $S_{w'}$ of a warrant w', since there are two signatures on a warrant: First, the original signer uses the signature scheme in Section 3 to sign the warrant, and then, the proxy signer will use BLS short signature scheme to sign it, these two signature schemes are secure. On the other hand, the above delegation does not require the secure channel for the delivery of the signed warrant by the original signer, *i.e.*, the original signer can publish w and So_w. More precisely, any adversary can get the original signer's signature on warrant w. Even this, the adversary cannot get the S_w of the proxy signer, because S_w should satisfy $e(H(w)P + PK_o, S_w) = e(PK_p, PK_p)$, and $e(H(w)P + PK_o, So_w) = e(P, PK_p)$. From P, So_w and PK_p to get S_w, this is CDHP.

The above delegation is a partial delegation with warrant [15]. It is can be regarded as the generation of the proxy key in proxy signature. The proxy secret key is S_w, and the proxy public key is $PK_o + PK_p$. Then the proxy signer can uses any ID-based signature schemes and ID-based blind signature schemes from pairings (takes the ID public key as $H_2(w)$) and secret key as S_w, the public key of PKG as $PK_o + PK_p$) to get proxy signature and proxy blind signature schemes.

Next, we give two applications of above delegation method in proxy signature: designing proxy signature scheme and a proxy blind signature scheme. We only describe the schemes without security analysis.

A Proxy Signature Scheme

Proxy signatures are very useful tools when one needs to delegate his/her signing capability to other party[15, 16]. Using above delegation, we give a new proxy signature scheme.

Setup: Define another cryptographic hash function: $H_1 : \{0,1\}^* \times \mathbb{G}_1 \to Z_q^*$. The system parameters $params = \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, H, H_1\}$, the original signer has public-secret key pair (PK_o, s_o), the proxy signer has public-secret key pair (PK_p, s_p).

Generation of the proxy key: The proxy signer receives a proxy key S_w using above delegation protocol.

Signing: For a message m, choose a random number $r \in \mathbb{Z}_q^*$, compute $U = r \cdot (H(w)P + PK_o)$. Compute $h = H_1(m||U)$ and $V = (h+r)^{-1}S_w$. The proxy signature on m is (U, V, w).
Verification: Verify that

$$e(U + H_1(m||U)(H(w)P + PK_o), V) = e(PK_p, PK_p).$$

A Proxy Blind Signature Scheme

Proxy blind signature is considered to be the combination of proxy signature and blind signature, so, it satisfies the security properties of both the blind signature and the proxy signature. Such signature is suitable for many applications where the users' privacy and proxy signature are required. Now, we give a new proxy blind signature scheme.

Setup: Same as above proxy signature scheme.
Generation of the proxy key: The proxy signer receives a proxy key S_w.
Proxy blind signature generation: Suppose that m is the message to be signed.

– The proxy signer randomly chooses a number $r \in_R \mathbb{Z}_q^*$, computes $U = r \cdot (H(w)P + PK_o)$, and sends U and the warrant w to the user.
– (Blinding) The user randomly chooses $\alpha, \beta \in_R Z_q^*$ as blinding factors. He/She computes $U' = \alpha U + \alpha\beta(H(w)P + PK_o)$ and $h = \alpha^{-1}H_1(m||U') + \beta$, sends h to the signer.
– (Signing) The signer sends back V, where $V = (r+h)^{-1}S_w$.
– (Unblinding) The user computes $V' = \alpha^{-1}V$ and outputs (m, U', V').

Then (U', V', w) is the proxy blind signature of the message m.
Verification: A verifier accepts this proxy blind signature if and only if

$$e(U' + H_1(m||U')(H(w)P + PK_o), V') = e(PK_p, PK_p).$$

7 Conclusion

In this paper, we proposed a new short signature scheme that is more efficient than BLS scheme. The security of this signature scheme depends on a new problem, namely k-CAA or $k + 1EP$. It is shown that $k + 1EP$ is no harder than the CDHP. Based on this basic signature scheme, a ring signature scheme and a new method for delegation are proposed.

Acknowledgements

We would like to thank Ben Lynn, Yi Mu and Xiaofeng Chen for their valuable discussions and comments on this work.

References

1. P.S.L.M. Barreto and H.Y. Kim, *Fast hashing onto elliptic curves over fields of characteristic 3*, Cryptology ePrint Archive, Report 2001/098, available at http://eprint.iacr.org/2001/098/.
2. P.S.L.M. Barreto, H.Y. Kim, B.Lynn, and M.Scott, *Efficient algorithms for pairing-based cryptosystems*, Advances in Cryptology-Crypto 2002, LNCS 2442, pp.354-368, Springer-Verlag, 2002.
3. P.S.L.M. Barreto, B.Lynn, and M.Scott, *On the selection of pairing-friendly groups*, SAC 2003. Lecture Notes in Computer Science, Springer- Verlag, 2003.
4. M. Bellare and P. Rogaway, *The exact security of digital signatures - How to sign with RSA and Rabin.* Proceedings of Eurocrypt'96, LNCS vol. 1070, Springer-Verlag, 1996, pp. 399-416.
5. A. Boldyreva, *Efficient threshold signature, multisignature and blind signature schemes based on the Gap-Diffie-Hellman -group signature scheme*, PKC 2003, LNCS 2139, pp.31-46, Springer-Verlag, 2003.
6. D. Boneh and M. Franklin, *Identity-based encryption from the Weil pairing*, Advances in Cryptology-Crypto 2001, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
7. D. Boneh, C. Gentry, B. Lynn and H. Shacham, *Aggregate and verifiably encrypted signatures from bilinear maps*, Eurocrypt 2003, LNCS 2656, pp.272-293, Springer-Verlag, 2003.
8. D. Boneh, B. Lynn, and H. Shacham, *Short signatures from the Weil pairing*, In C. Boyd, editor, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.514-532, Springer-Verlag, 2001.
9. J.C. Cha and J.H. Cheon, *An identity-based signature from gap Diffie-Hellman groups*, Public Key Cryptography - PKC 2003, LNCS 2139, pp.18-30, Springer-Verlag, 2003.
10. N.T. Courtois, M. Finiasz and N. Sendrier, *How to achieve a McEliece-based Digital Signature Schem*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.157-174, Springer-Verlag, 2001.
11. S. D. Galbraith, K. Harrison, and D. Soldera, *Implementing the Tate pairing*, ANTS 2002, LNCS 2369, pp.324-337, Springer-Verlag, 2002.
12. S. Goldwasser, S. Micali and R. Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*, SIAM Journal of computing, 17(2), pp. 281-308, April 1988.
13. F. Hess, *Efficient identity based signature schemes based on pairings*, SAC 2002, LNCS 2595, pp.310-324, Springer-Verlag, 2002.
14. A. Joux, *A one round protocol for tripartite Diffie-Hellman*, ANTS IV, LNCS 1838, pp.385-394, Springer-Verlag, 2000.
15. S. Kim, S. Park, and D. Won, *Proxy signatures, revisited*, In Pro. of ICICS 97, LNCS 1334, Springer-Verlag, pp. 223-232, 1997.
16. M. Mambo, K. Usuda, and E. Okamoto, *Proxy signature: Delegation of the power to sign messages*, In IEICE Trans. Fundamentals, Vol. E79-A, No. 9, Sep., pp. 1338-1353, 1996.
17. U. Maurer, *Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms*, Advances in Cryptology-Crypto 94, LNCS 839, pp.271-281, Springer-Verlag, 1994.
18. S. Mitsunari, R. Sakai and M. Kasahara, *A new traitor tracing*, IEICE Trans. Vol. E85-A, No.2, pp.481-484, 2002.

19. J. Patarin, N. Courtois and L. Goubin, *QUARTZ, 128-bit long digital signatures*, CT-RSA 2001, LNCS 2020, pp. 282-297, Springer-Verlag, 2001.
20. K.G. Paterson, *ID-based signatures from pairings on elliptic curves*, Electron. Lett., Vol.38, No.18, pp.1025-1026, 2002.
21. R.L. Rivest, A. Shamir and Y. Tauman, *How to leak a secret*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.552-565, Springer-Verlag, 2001.
22. A.R. Sadeghi and M. Steiner,*Assumptions related to discrete logarithms: why subtleties make a real difference*, Eurocrypt 2001, LNCS 2045, pp.243-260, Springer-Verlag, 2001.
23. R. Sakai, K. Ohgishi and M. Kasahara, *Cryptosystems based on pairing*, SCIS 2000-C20, Jan. 2000. Okinawa, Japan.
24. R. Sakai and M. Kasahara, *Cryptosystems based on pairing over elliptic curve*, SCIS 2003, 8C-1, Jan. 2003. Japan.
25. R. Sakai and M. Kasahara, *ID based cryptosystems with pairing on elliptic curve*, Cryptology ePrint Archive, Report 2003/054, available at http://eprint.iacr.org/2003/054/.
26. A. Shamir, *Identity-based cryptosystems and signature schemes*, Advances in Cryptology-Crypto 84, LNCS 196, pp.47-53, Springer-Verlag, 1984.
27. N.P. Smart, *An identity based authenticated key agreement protocol based on the Weil pairing*, Electron. Lett., Vol.38, No.13, pp.630-632, 2002.
28. F. Zhang and K. Kim, *ID-based blind signature and ring signature from pairings*, Proc. of Asiacrpt2002, LNCS 2501, pp. 533-547, Springer-Verlag, 2002.

Appendix A: Proof of Theorem 2

Proof. $BDHP \Rightarrow BSDHP$ is trivial.

$BSDHP \Rightarrow BIDHP:$

Given P, aP, cP, set the input of BSDHP to be

$$Q = aP, Q_1 = P = a^{-1}Q, Q_2 = cP = ca^{-1}Q,$$

then BSDHP outputs

$$e(Q,Q)^{(a^{-1})^2ca^{-1}} = e(aP,aP)^{(a^{-1})^2ca^{-1}} = e(P,P)^{a^{-1}c}.$$

$BIDHP \Rightarrow BSDHP:$

Given P, aP, cP, set the input of BIDHP to be

$$Q = aP, Q_1 = P = a^{-1}Q, Q_2 = cP = ca^{-1}Q,$$

then BIDHP outputs

$$e(Q,Q)^{(a^{-1})^{-1}ca^{-1}} = e(aP,aP)^c = e(P,P)^{a^2c}.$$

$BSDHP \Rightarrow BDHP:$

Given P, aP, bP, cP, set the input of BSDHP to be $< P, aP, cP >$, $< P, bP, cP >$, and $< P, aP + bP, cP >$, respectively, then we have $e(P,P)^{a^2c}$, $e(P,P)^{b^2c}$ and $e(P,P)^{(a+b)^2c}$, so we get

$$e(P,P)^{abc} = (e(P,P)^{(a+b)^2c}/(e(P,P)^{a^2c} \cdot e(P,P)^{b^2c}))^{1/2}.$$

□