Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings

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. Verifiably encrypted signatures are used when Alice wants to sign a message for Bob but does not want Bob to possess her signature on the message until a later date. Such signatures are used in optimistic contact signing to provide fair exchange. Partially blind signature schemes are an extension of blind signature schemes that allows a signer to sign a partially blinded message that include pre-agreed information such as expiry date or collateral conditions in unblinded form. These signatures are used in applications such as electronic cash (e-cash) where the signer requires part of the message to be of certain form. In this paper, we propose a new verifiably encrypted signature scheme and a partially blind signature scheme, both based on bilinear pairings. We analyze the security and efficiency of these schemes and show that they are more efficient than the previous schemes of their kinds.

Keywords: Verifiably encrypted signature, partially blind signature, Bilinear pairings.

1 Introduction

When Alice wants to sign a message for Bob but does not want Bob to possess her signature on the message immediately. Alice can achieve this by encrypting her signature using the public key of a trusted third party (adjudicator), and sending the result to Bob along with a proof that she has given him a valid encryption of her signature. Bob can verify that Alice has signed the message but cannot deduce any information about her signature. At a later stage, Bob can either obtain the signature from Alice or resort to the adjudicator who can reveal Alice's signature. There are many applications of such verifiably encrypted signature scheme, such as online contract signing [3, 4]. Boneh *et al.* [10] gave a verifiably encrypted signature scheme as an application of their aggregate signature. Their scheme is based on a short signature due to Boneh, Lynn, and Shacham (BLS) [11] constructed from bilinear pairings.

Blind signatures were first introduced by Chaum [13] and play a central role in cryptographic protocols such as e-cash or e-voting that require user anonymity. However, when we use blind signature to design e-cash schemes, there are two obvious shortcomings: (1) To prevent a customer from double-spending his e-cash, the bank has to keep a database which stores all spent e-cash to check whether a specified e-cash has been spent or not by searching this database. This operation is referred to as the freshness checking (or the

double-spending checking) of e-cash. Certainly, the database kept by the bank may grow unlimitedly. (2) The bank cannot inscribe the value on the blindly issued e-cash. To believe the face value of e-cash, there are two conventional solutions: First, the bank uses different public keys for different coin values. In this case, the shops and customers must always carry a list of those public keys in their electronic wallet, which is typically a smart card whose memory is very limited. Second solution, the bank can use the cut-and-choose algorithm [13] in the withdraw phase. But this is very inefficient.

Partially blind signatures were introduced by Abe and Fujisaki [1] to allow the signer to explicitly include some agreed information in the blind signature. Using partially blind signatures in e-cash system, we can prevent the bank's database from growing unlimitedly. Because the bank assures that each e-cash issued by it contains the information it desires, such as the date information. By embedding an expiration date into each e-cash issued by the bank, all expired e-cash recorded in the bank's database can be removed. At the same time, each e-cash can be embedded the face value, the bank can know the value on the blindly issued e-cash. A number of partially blind signature schemes using different assumptions have been proposed. Abe and Fujisaki's scheme is based on RSA [1]. Abe and Okamoto's scheme is based on discrete logarithm problem [2] and Fan and Lei's scheme is based on quadratic residues problem [15].

In this paper, we propose a new verifiably encrypted signature scheme and a partially blind signature scheme, both based on bilinear pairings. We analyze security of these schemes and show that they are more efficient than previous schemes.

The rest of the paper is organized as follows. In the next section we give a brief introduction to bilinear pairings and describe two signature schemes from bilinear pairings. Section 3 gives the definition and security properties of verifiably encrypted signature schemes and partially blind signature schemes. In Section 4 we describe our proposed verifiably encrypted signature schemes and in Section 5 analye its security. Sections 6 and 7 give our proposed partially blind signature scheme and its analysis, respectively. Section 8 concludes the paper.

2 Preliminaries

In recent years, bilinear pairings have been used to construct numerous new cryptographic primitives [9, 11, 12, 17, 18, 20, 22–26]. We recall the basic concept and properties of bilinear pairings.

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 bilinear pairing with the following properties:

1. **Bilinearity:** $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1, a, b \in Z_q$
2. **Non-degeneracy:** There exists $P, Q \in G_1$ 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 $P, Q \in G_1$.

Three well-known problems in groups that is commonly used in Cryptography are, Discrete Logarithm Problem (DLP), Decision Diffie-Hellman Problem (DDHP)and Computational Diffie-Hellman Problem (CDHP). For the sake of brevity we do not state the problems here and refer the reader to [8, 19]. Two variations of CDHP are:

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

Generalizing these two problems, we can obtain the following problems:

Definition 1 (k-wCDHP (k-weak Computational Diffie-Hellman Problem)[20]). *Given $k+1$ values $< P, yP, y^2P, \ldots, y^kP >$, compute $\frac{1}{y}P$.*

Definition 2 (k+1 Exponent Problem [27]). *Given $k+1$ values $< P, yP, y^2P, \ldots, y^kP >$, compute $y^{k+1}P$.*

The following theorem due to [27], gives the relationship between the two problems:

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

Assumptions: We assume that DLP, CDHP, Inv-CDHP, Squ-CDHP and k+1 Exponent Problem are hard, which mean there are no polynomial time algorithm to solve them with non-negligible probability.

When the DDHP is easy but the CDHP is hard on the group G, we call G a *Gap Diffie-Hellman* (*GDH*) *group*. 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. More details can be found in [9, 12, 17].

Schemes in this paper can work on any GDH group. Throughout this paper, we define the system parameters in all schemes are as follows: Let P be a generator of $\mathbb{G}_1$ with order q, the bilinear pairing is given by $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. These system parameter can be obtained using a **GDH Parameter Generator** $\mathcal{IG}$ [9]. Define two cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$, in general, $|q| \geq \lambda \geq 160$, and $H_0 : \{0,1\}^* \to \mathbb{G}_1^*$. Denote $params = \{\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, H, H_0\}$.

2.1 The Basic Signature Scheme

A signature scheme is described by the following four algorithms : a parameter generation algorithm Generate, a key generation algorithm KeyGenparam, a signature generation algorithm Sign and a signature verification algorithm Ver.

We recall a basic signature scheme from bilinear pairings proposed in [27].

1. Generate. Generate the system parameters: *params*.
2. KeyGenparam. Pick random $x \in_R Z_q^*$, and compute $P_{pub} = xP$. The public key is P_{pub}. The secret key is x.
3. Sign. Given a secret key x, and a message m. Compute $S = \frac{1}{H(m)+x}P$.
4. Ver. Given a public key P_{pub}, a message m, and a signature S, verify if $e(H(m)P + P_{pub}, S) = e(P, P)$.

This signature scheme was proposed at [27], it can be regarded as being derived from Sakai-Kasahara's new ID-based encryption scheme with pairing [23]. In [27], the authors proved that this signature scheme was secure against existential forgery on adaptive chosen-message attacks (in the random oracle model) assuming the "$k+1$ Exponent Problem" is hard in $\mathbb{G}_1$.

2.2 The Blind GDH Signature

We introduce a blind GDH signature scheme as follows:

- Generate. Generate the system parameters: $params$.
- KeyGenparam. Pick random $x \in_R Z_q^*$, and compute $P_{pub} = xP$. The public key is P_{pub}. The secret key is x.
- Blind signature issuing. The user wants a message $m \in \{0,1\}^*$ to be signed.
  - (Blinding) The user randomly chooses a number $r \in_R Z_q^*$, computes $M' = H_0(m) + rP$, and sends M' to the signer.
  - (Signing) The signer sends back σ', where $\sigma' = x \cdot M'$.
  - (Unblinding) The user then computes the signature $\sigma = \sigma' - rP_{pub}$ and outputs (m, σ).
- Ver. Given a public key P_{pub}, a message m, and a signature σ, verify if $e(P_{pub},\ H_0(m)) = e(P,\ \sigma)$ holds.

This blind signature scheme can be regarded as the blind version of BLS signature scheme [11], that was firstly mentioned in [25]. In [7], Boldyreva gave a security proof of this blind signature scheme, they showed that this blind signature scheme was secure against one-more forgery under the "Chosen-target CDH" [7] assumption.

3 Definitions

In this section, we introduce the definitions and security properties of verifiably encrypted signature and partially blind signature.

Definition 3 (Verifiably Encrypted Signature [10]). *A verifiably encrypted signature scheme consists of three entities: signer, verifier and adjudicator. There are seven algorithms. Three,* **KeyGen**, **Sign**, *and* **Verify**, *are analogous to those in ordinary signature schemes. The others,* **AdjKeyGen**, **VESigCreate**, **VESigVerify**, *and* **Adjudicate**, *provide the verifiably encrypted signature capability.*

- **KeyGen, Sign, Verify:** *These are key generation, signing and verification of the signer, they are same as in standard signature schemes.*
- **AdjKeyGen:** *This is generating a public-private key pair (APK, ASK) for the adjudicator.*
- **VESigCreate:** *Given a secret key SK, a message m, and an adjudicator public key APK, compute a verifiably encrypted signature ν on m.*
- **VESigVerify:** *Given a public key PK, a message m, an adjudicator public key APK, and a verifiably encrypted signature ν, verify that ν is a valid verifiably encrypted signature on m under key PK.*
- **Adjudicate:** *Given an adjudicator keypair (APK, ASK), a certified public key PK, and a verifiably encrypted signature σ on some message m, extract and output ν, an ordinary signature on m under PK.*

Besides the ordinary notions of signature security in the signature component, we require three security properties of verifiably encrypted signatures:

Validity: This requires that **VESigVerify**(m,**VESigCreate**(m)) and **Verify**(m, **Adjudicate** (**VESigCreate**(m))) hold for all m and for all properly-generated keypairs and adjudicator keypairs.

Unforgeability: This requires that it be difficult to forge a valid verifiably encrypted signature.

Opacity: This requires that it be difficult, given a verifiably encrypted signature, to extract an ordinary signature on the same message.

Partially blind signatures were introduced by Abe and Fujisaki [1]. In [2], Abe and Okamoto presented a formal definition of partially blind signature schemes. The following definition is based on Abe-Okamoto's definition.

Definition 4 (Partially Blind Signature). *A Partially blind signature scheme consists of three participants: signer, user and verifier. There are three algorithms:* **Key Generation** *algorithm,* **Partially blind signature issuing** *algorithm and* **Verification** *algorithm.*

- **Key Generation** *is a probabilistic polynomial-time algorithm that takes security parameter k and outputs a public and secret key pair (pk, sk).*
- **Partially blind signature issuing** *is a interactive protocol between the signer and the user. The public input of the user contains pk and the public information info. The public input of the signer contains the public information info. The private input tape of the the signer contains sk, and that for the user contains message m. When they stop,the public output of the user contains either completed or not completed, the private output of the user contains either "fail" or $(\mathsf{info}, m, \sigma)$.*
- **Verification** *is a (probabilistic) polynomial-time algorithm that takes (pk, info, m, σ) and outputs either accept or reject.*

Security of a partially blind signature scheme is in terms of three requirements: *completeness*, *partial blindness* and *non-forgeability*. Partial blindness must satisfy the following two properties: (1). The signer assures that an issued signature contains the information that it desires, and none can remove the embedded information from the signature. (2). For the same embedded information, the signer cannot link a signature to the instance of the signing protocol that produces the corresponding blind signature. The most powerful attack on a blind signature is *one-more signature forgery* introduced by Pointcheval and Stern in [21]. A partially blind signature scheme is called unforgeable against one-more forgery under chosen message attack, that means for each info, for some integer l, there is no probabilistic polynomial-time adversary $\mathcal{A}$ that can compute, after l interactions with the signer, $l + 1$ signatures with non-negligible probability. A partially blind signature scheme is called secure if it satisfies these requirements.

4 A New Verifiably Encrypted Signature Scheme

At Eurocrypt 2003, Boneh, Gentry, Lynn and Shacham [10] proposed a verifiably encrypted signature scheme as an application of their aggregate signatures. Their scheme is based on

a short signature due to Boneh, Lynn, and Shacham (BLS) [11] constructed from bilinear pairings. To get the verifiably encrypted signature, they used the ElGamal encryption algorithm. In this section, we propose a new verifiably encrypted signature scheme from bilinear pairings. This new scheme does not require the ElGamal encryption.

The new verifiably encrypted signature scheme uses the basic signature scheme in 2.1 and works as follows.

Key Generation. KeyGen and **AdjKeyGen** are the same as the key generation algorithm in the basic signature scheme, i.e., the system parameters are $\{\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, H\}$, the signer and adjudicator have the public-secret key pair (P_{pub}, x) and (P_{pubAd}, x_a), respectively.

Signing, Verification. $Sign$ and $Verify$ are the same as in the basic signature scheme, i.e., for a message m, the signature is $\sigma = \frac{1}{H(m)+x}P$, the verification is $e(H(m)P + P_{pub},\ \sigma) = e(P,\ P)$.

VESig Creation. Given a secret key $x \in Z_p$ a message m, and an adjudicator's public key P_{pubAd}, compute $\nu = \frac{1}{H(m)+x}P_{pubAd}$. The verifiably encrypted signature for message m is ν.

VESig Verification. Given a public key P_{pub}, a message m, an adjudicator's public key P_{pubAd}, and a verifiably encrypted signature ν, accept ν if and only if the following equation holds:

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

Adjudication. Given an adjudicator's public key P_{pubAd} and corresponding private key $x_a \in Z_q$, a certified public key P_{pub}, and a verifiably encrypted signature ν on some message m, ensure that the verifiably encrypted signature is valid; then output $\sigma = x_a^{-1}\nu$.

5 Analysis of the Verifiably Encrypted Signature Scheme

5.1 Security

We show that the proposed signature scheme satisfies properties of a verifiably encrypted signature scheme.

Validity. ν is the verifiably encrypted signature for message m. Since we have

$$e(H(m)P + P_{pub},\ \nu) = e((H(m)+x)P,\ \frac{1}{H(m)+x}P_{pubAd}) = e(P,\ P_{pubAd}),$$

this means **VESigVerify**(m,**VESigCreate**(m)) holds, and

$$e(H(m)P + P_{pub},\ x_a^{-1}\nu) = e((H(m)+x)P,\ x_a^{-1} \cdot \frac{1}{H(m)+x} \cdot x_a P) = e(P, P)$$

this means **Verify**(m, **Adjudicate**(**VESigCreate**(m))), so *Validity* holds.

Unforgeability: For the unforgeability, we have the following claim:

Claim 1. If the basic signature scheme is secure against existential forgery, then the new verifiably encrypted signature scheme is secure against existential forgery.

To prove this claim, we show that if the new verifiably encrypted signature scheme is forgeable against existential forgery, then the basic signature scheme is forgebale too. That is if there is a probabilistic polynomial time forger algorithm $\mathcal{F}$ with a non-negligible probability ϵ under an adaptive chosen message attack for the verifiably encrypted signature scheme, then using $\mathcal{F}$, we can construct a new probabilistic polynomial time forger algorithm $\mathcal{F}'$ such that $\mathcal{F}'$ can forge a signature of the basic signature scheme with non-negligible probability. Because the basic signature scheme is secure against existential forgery using adaptive chosen-message attacks (in the random oracle model) and assuming the $k+1$ Exponent Problem [27] is hard in $\mathbb{G}_1$, then the new verifiably encrypted signature scheme is unforgeable.

We adopt the security model of Boneh et al. [10]. We assume that the basic signature scheme is given as in 2.1 : $\{\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, H\}$, and the public-secret key pair of the signer is (P_{pub}, x). The forger algorithm $\mathcal{F}'$ sets up a verifiably encrypted signature scheme $\mathcal{V}$ based on the basic signature scheme: $\mathcal{F}'$ generates a key, $(x_0, P_0) \leftarrow$ **KeyGen**, which serves as the adjudicator's key. Suppose a probabilistic polynomial time forger algorithm $\mathcal{F}$ for the verifiably encrypted signature scheme $\mathcal{V}$ is given. Now, $\mathcal{F}'$ runs $\mathcal{F}$ on $\mathcal{V}$ and if $\mathcal{F}$ generates a forged verifiably encrypted signature ν' for a message m', then $\mathcal{F}'$ produces a forged signature σ' of the basic signature scheme for this message m', where $\sigma' = x_0^{-1}\nu'$.

Opacity: For the opacity, we have the following claim:

Claim 2. If the basic signature scheme is secure against existential forgery and the DLP is hard, then the new verifiably encrypted signature scheme is secure against extraction.

Suppose given a verifiably encrypted signature ν for a message m, an adversary $\mathcal{A}$ wants to compute the signature σ of the signer on the message m. $\mathcal{A}$ either forge a signature of the signer for message m (under the signer's public key P_{pub}), directly, or extract a signature σ' from ν, such that $e(H(m)P + P_{pub}, \sigma') = e(P, P)$.

Assuming that the basic signature scheme is secure against existential forgery, then it is impossible to forge a signature of the signer for message m. We show that extracting a signature σ' from ν such that $e(H(m)P + P_{pub}, \sigma') = e(P, P)$, is equivalent to solving DLP. Since ν satisfies

$$e(H(m)P + P_{pub},\ \nu) = e(P,\ P_{pubAd}) = e(P, P)^{x_a},$$

and due to the *bilinearity* property of the pairing, we have

$$e(H(m)P + P_{pub},\ x_a^{-1}\nu) = e(P,\ P).$$

Due to the $non-degeneracy$ of bilinear pairing, we have $\sigma' = x_a^{-1}\nu$. So, to get σ', the adversary $\mathcal{A}$ should know x_a which is the discrete logarithm of P_{pubAd} in base P.

5.2 Efficiency

We compare our verifiably encrypted signature scheme with Boneh et al.'s scheme [10] from the view point of computation overhead. We denote Pa the pairing operation, Pm

the point scalar multiplication on $\mathbb{G}_1$, Ad the point addition on $\mathbb{G}_1$, Mu the multiplication on $\mathbb{G}_2$, Inv the inversion in Z_q and MTP the MapToPoint hash operation in BLS scheme [11]. We summarize the result in Table 1(we ignore the general hash operation).

$Schemes$	$VESig\ Creation$	$VESig\ Verification$	$Adjudication$
$Proposed$	$1Inv + 1Pm$	$2(or\ 1)Pa + 1Pm + 1Ad$	$1Inv + 1Pm$
$Boneh\ et\ al.'s$	$1MTP + 3Pm + 1Ad$	$1MTP + 3Pa + 1Mu$	$1Pm + 1Ad$

Table 1. Comparison of our scheme and the Boneh et al.'s scheme

We note that the computation of the pairing is the most time-consuming. Although there have been many papers discussing the complexity of pairings and how to speed up the pairing computation [5, 6, 16], the computation of the pairing still remains time-consuming. In our scheme, we can precompute $e(P, P_{pubAd})$ and publish it as part of the adjudicator's public keys, therefore, there is only one pairing operation in **VESig verification**, but there are three pairing operations in Boneh et al.'s scheme. On the other hand, our scheme does not require special hash functions but a general cryptographic hash functions such as SHA-1 or MD5. In Boneh et al.'s scheme, there is a special hash operation: MapToPoint, there is at least one quadratic or cubic equation over finite field need to be solved. Hence, our verifiably encrypted signature scheme is much more efficient than Boneh et al.'s scheme. Finally, our signature is shorter than Boneh et al.'s signature and consists of one element of $\mathbb{G}_1$ in our scheme, but two elements in Boneh et al.'s scheme.

6 A New Partially Blind Signature Scheme

We propose a new partially blind signature scheme from bilinear pairings. The proposed scheme can be regarded as the combination of the basic signature scheme in 2.1 and the blind GDH signature in 2.2.

The system parameters are: $params = \{\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, H, H_0\}$.

[**Key generation:**]
The signer picks random $x \in_R Z_q^*$, and computes $P_{pub} = xP$. The public key is P_{pub}. The secret key is x.

[**Partially blind signature issuing protocol:**]
Suppose that m be the message to be signed and c be the public information. The protocol is shown in Fig. 1.

- (Generation of the public information) The user and signer generate the public information c together.
- (Blinding) The user randomly chooses a number $r \in_R Z_q^*$, computes $U = H_0(m||c) + r \cdot (H(c)P + P_{pub})$[1], and sends U to the signer.

[1] In the original partially blind signature scheme of INDOCRYPT 2003 publish version, $U = rH_0(m||c)$, and in the Unblinding phase, $S = r_1V$. As mentioned by Sherman S.M. Chow et al.[14], it does not meet the unlinkability. The randomness introduced during the blinding phase can be removed easily by the bilinearity of the pairings operation, i.e., after obtain the partially blind signature (S, m, c) and the instance of the issue protocol (U, V), any party can check if $e(S, U) = e(V, H_0(m||c))$ to trace a user or a partially blind signature.

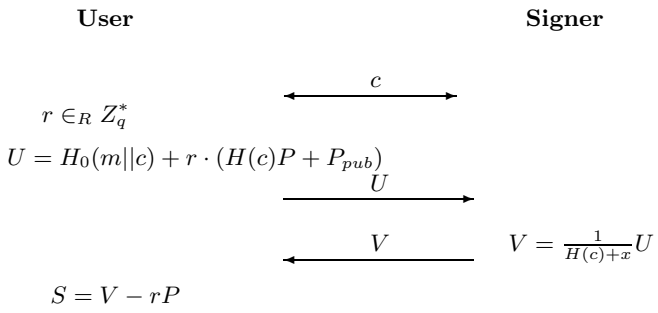

Fig. 1. The partially blind signature issuing protocol

- (Signing) The signer sends back V, where $V = (H(c) + x)^{-1}U$.
- (Unblinding) The user computes $S = V - rP$.

Then (S, m, c) is the partially blind signature of the message m and public information c.

[**Verification:**]
A verifier can accept this partially blind signature if and only if

$$e(H(c)P + P_{pub},\ S) = e(P,\ H_0(m||c)).$$

7 Analysis of the New Partially Blind Signature Scheme

7.1 Completeness

The completeness can be justified by the following equations:

$$\begin{aligned}
& e(H(c)P + P_{pub},\ S) \\
&= e((H(c) + x)P,\ V - rP) \\
&= e((H(c) + x)P,\ (H(c) + x)^{-1}U - rP) \\
&= e((H(c) + x)P,\ (H(c) + x)^{-1}U)e((H(c) + x)P,\ -rP) \\
&= e(P,\ H_0(m||c) + r \cdot (H(c)P + P_{pub})e(H(c)P + P_{pub},\ -rP) \\
&= e(P,\ H_0(m||c))e(P,\ r \cdot (H(c)P + P_{pub})e(H(c)P + P_{pub},\ -rP) \\
&= e(P,\ H_0(m||c))
\end{aligned}$$

7.2 Partial Blindness

In the Blinding phase, r is chosen randomly from $\mathbb{Z}_q^*$ and so $H_0(m||c) + r \cdot (H(c)P + P_{pub})$ is a random element of the group $\mathbb{G}_1$. The signer receives this random information and the public information which he already knows and so no information about the message will be leaked.

The signer is assured that a signature issued by him contains the public information that he has agreed on and this information cannot be removed from the signature. This is

true because if a malicious user could generate c' and replace c from the signer's signature (S, m, c) to obtain a signature with c'. Then we have

$$e(H(c')P + P_{pub},\ S) = e(P,\ H_0(m||c')),$$

that means

$$e((H(c') - H(c))P,\ S) = e(P,\ H_0(m||c') - H_0(m||c)),$$

so, c and c' should satisfy $(H(c')-H(c))S = H_0(m||c')-H_0(m||c)$. This is unlikely, because H, H_0 are cryptographic hash functions.

Given a valid signature (S, m, c) and any view (U, V), there always exists a unique blinding factor r such that $V - S = rP$. So, due to the randomness of blinding factor chosen during the Blinding phase and the fact that the public information is independent of the message, even if the same embedded information be used for two messages, the signer cannot link a signature to the corresponding instance of signature issuing protocol.

Hence the partially blind signature scheme satisfies the partial blindness property.

7.3 Unforgeability

To show that the proposed partially blind signature scheme is unforgeable, we first transform it to a fully blind signature scheme and then prove that the fully blind signature scheme is unforgeable.

For any public information c, the signer sets up the system parameters and public key as: $params = \{\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, H, H_0\}$ and Q. Here $Q = H(c)P + xP = sP$, $x \in_R Z_q^*$. The secret key is $s^{-1} = (H(c) + x)^{-1}$. Let m be the message to be signed. The blind signature issuing protocol of this fully blind signature scheme is shown as follows:

- (Blinding) The user randomly chooses a number $r \in_R Z_q^*$, computes $U = H_0(m) + rQ$, and sends U to the signer.
- (Signing) The signer sends back V, where $V = s^{-1}U = (H(c) + x)^{-1}U$.
- (Unblinding) The user computes $S = V - rP$.

The verification this blind signature is

$$e(Q,\ S) = e(P,\ H_0(m)).$$

We call above fully blind signature scheme FuBS. FuBS is derived from the proposed partially blind signature scheme. It is easy to see that if a massage-signature pair (m, c, S) can be forged for the proposed partially blind signature scheme, then a blind signature on the message $m' = m||c$ for the corresponding FuBS can be forged. So, we have the following lemma.

Lemma 1. *If FuBS is secure against one-more forgery under chosen message attack. Then the proposed partially blind signature scheme is secure against one-more forgery under chosen message attack.*

Next, we show that FuBS is secure against one-more forgery under chosen message attack. It is easy to see that FuBS is very similar to the blind GDH signature in section 2.2. We will

use similar technique in [7], where the author defined "Chosen target CDH" assumption and proved that their blind signature scheme is secure assuming the hardness of the chosen-target CDH problem. First, we give a variations of chosen-target CDH problem, named "Chosen target Inverse CDH" problem. We propose the problem and assumption as follows:

Definition 5. *Let $\mathbb{G}_1$ be* GDH *group of prime order q and P is a generator of $\mathbb{G}_1$. Let s be a random element of $\mathbb{Z}_q^*$ and $Q = sP$. Let $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1$ be a cryptographic hash function. The adversary $\mathcal{A}$ is given input (q, P, Q, H_0) and has access to the target oracle $T_{\mathbb{G}_1}$ that returns a random point U_i in $\mathbb{G}_1$ and the helper oracle* `Inv-cdh-`$s(\cdot)$ *(compute $s^{-1} \cdot (\cdot)$). Let q_T and q_H be the number of queries $\mathcal{A}$ made to the target oracle and the helper oracle respectively. The advantage of the adversary attacking the chosen-target inverse CDH problem $\mathsf{Adv}_{\mathbb{G}_1}^{ct-icdh}(\mathcal{A})$ is defined as the probability of $\mathcal{A}$ to output a set of l pairs $((V_1, j_1), (V_2, j_2), \ldots, (V_l, j_l))$, for all $i = 1, 2, \ldots, l$ $\exists$ $j_i = 1, 2, \ldots, q_T$ such that $V_i = s^{-1}U_{j_i}$ where all V_i are distinct and $q_H < q_T$.*
The chosen-target inverse CDH assumption states that there is no polynomial-time adversary $\mathcal{A}$ with non-negligible $\mathsf{Adv}_{\mathbb{G}_1}^{ct-icdh}(\mathcal{A})$.

The following theorem shows that FuBS is secure assuming the chosen-target inverse CDH problem is hard.

Theorem 2. *If the chosen-target inverse CDH assumption is true in the group $\mathbb{G}_1$ then FuBS is secure against one-more forgery under chosen message attack.*

Proof. (sketch). If there is a probabilistic polynomial time one-more forger algorithm $\mathcal{F}$ with a non-negligible probability ϵ for FuBS under an chosen message attack, then using $\mathcal{F}$, we can construct an algorithm $\mathcal{A}$ such that $\mathcal{A}$ can solve the chosen-target inverse CDH problem with a non-negligible probability.

Suppose that a probabilistic polynomial time forger algorithm $\mathcal{F}$ is given. Suppose that $\mathcal{A}$ is given a challenge as in Definition 5. Now $\mathcal{F}$ has access to a blind signing oracle $s(\cdot)$ and the random hash oracle $H_0(\cdot)$. First, $\mathcal{A}$ provides $(\mathbb{G}_1, \mathbb{G}_2, e, q, P, H_0, Q)$ to $\mathcal{F}$ and $\mathcal{A}$ has to simulate the random hash oracle and the blind signing oracle for $\mathcal{F}$.

Each time $\mathcal{F}$ makes a new hash oracle query which differs from previous one, $\mathcal{A}$ will forward to its target oracle and returns the reply to $\mathcal{F}$. $\mathcal{A}$ stores the pair query-reply in the list of those pairs. If $\mathcal{F}$ makes a query to blind signing oracle, $\mathcal{A}$ will forward to its helper oracle `Inv-cdh-`$s(\cdot)$ and returns the answer to $\mathcal{F}$.

Eventually $\mathcal{F}$ halts and outputs a list of message-signature pairs $((m_1, S_1), (m_2, S_2), \ldots, (m_l, S_l))$. $\mathcal{A}$ can find m_i in the list stored hash oracle query-reply for $i = 1, 2, \ldots, l$. Let j_i be the index of the found pair, then $\mathcal{A}$ can output its list as $((S_1, j_1), (S_2, j_2), \ldots, (S_l, j_l))$. Then this list is a solution to the problem in Definition 5. □

From Lemma 1 and Theorem 2, we have the following theorem.

Theorem 3. *The proposed partially blind signature scheme is unforgeable under the chosen-target inverse CDH assumption in the group $\mathbb{G}_1$.*

7.4 Advantages

Comparing with previous partially blind signature schemes, such as [1], [2], [15], etc, the new partially blind signature scheme has a number of advantages:

A1. **Short signature**. In the proposed partially blind signature scheme, the signature only consists of an element in $\mathbb{G}_1$. 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 with compression techniques. So, like BLS short signature scheme [11], our signature scheme can provide the short signature, the signature length is half the size of a DSA signature for a similar level of security. Short signatures are needed in low-bandwidth communication environments. An important application of partially blind signature is in e-cash system. E-coins are stored in users' electronic wallets which are typically implemented in smart cards with limited memory. The short length of the proposed signature makes the system much more practical.

A2. **Efficient**. The scheme can be implemented using elliptic curve cryptosystem, and is very efficient from the view point of the user and the bank. In the partially blind signature issuing protocol, the user only needs to perform *1MTP*, *3Pm* and *2Ad*, the bank only needs to perform *1Inv* and *1Pm*. In the verification, two pairing operations are needed (As we noted in 5.2, the computation of the pairing is the most time-consuming). In e-cash systems the verification will be done by the shop that can be assumed to have more computation power.

A3. **Batch verify** (For the same public information c). The efficiency of the system is of paramount importance when the number of verifications is considerably large (*e.g.*, when a bank issues a large number of electronic coins and the customer wishes to verify the correctness of the coins). The proposed partially blind signature scheme is very efficient when we consider the batch verification for the same public information c. Assuming that $S_1, S_2, \cdots, S_n$ are partially blind signatures on messages $m_1, m_2, \cdots, m_n$ with the same public information c. The batch verification is then to test if the following equation holds:

$$e(H(c)P + P_{pub}, \sum S_i) = e(P, \sum H_0(m_i||c)).$$

8 Conclusion

Verifiably encrypted signature and partially blind signature are very important and useful cryptographic primitives. We proposed a new verifiably encrypted signature scheme and a partially blind signature scheme, both based on bilinear pairings. We analyzed the security and efficiency of them and showed that they are efficient than the previous schemes.

References

1. M. Abe and E. Fujisaki, *How to date blind signatures*, Advances in Cryptology - Asiacrypt 1996. LNCS 1163, pp. 244-251, Springer-Verlag, 2002.
2. M. Abe and T. Okamoto, *Provably secure partially blind signatures*, Advances in Cryptology - CRYPTO 2000. LNCS 1880, pp. 271-286, Springer-Verlag, 2000.

3. N. Asokan, V. Shoup and M. Waidner, *Optimistic fair exchange of digital signatures*. IEEE J. Selected Areas in Comm., 18(4):593-610, April 2000.
4. F. Bao, R. Deng and W. Mao. *Efficient and practical fair exchange protocols with offline TTP*. In Proceedings of IEEE Symposium on Security and Privacy, pp. 77-85, 1998.
5. 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.
6. P.S.L.M. Barreto, B.Lynn, and M.Scott, *On the selection of pairing-friendly groups*, to appear in the Workshop on Selected Areas in Cryptography (SAC) 2003.
7. A. Boldyreva, *Efficient threshold signature, multisignature and blind signature schemes based on the Gap-Diffie-Hellman -group signature scheme*, Public Key Cryptography - PKC 2003, LNCS 2139, pp.31-46, Springer-Verlag, 2003.
8. D. Boneh, *The decision Diffie-Hellman problem*, Proceedings of the Third Algorithmic Number Theory Symposium, LNCS 1423, pp. 48-63, Springer-Verlag, 1998.
9. 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.
10. 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.
11. 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.
12. 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.
13. D. Chaum, *Blind signatures for untraceable payments*, Advances in Cryptology-Crypto 82, Plenum, NY, pp.199-203, 1983.
14. Sherman S.M. Chow, and S.M. Yiu, personal communication, 2004.
15. C.I. Fan and C.L. Lei, *Low-computation partially blind signatures for electronic cash*, IEICE Transactions on Fundamentals ofElectronics, Communications and Computer Sciences E81-A(5) (1998) 818-824.
16. S. D. Galbraith, K. Harrison, and D. Soldera, *Implementing the Tate pairing*, ANTS 2002, LNCS 2369, pp.324-337, Springer-Verlag, 2002.
17. F. Hess, *Efficient identity based signature schemes based on pairings*, SAC 2002, LNCS 2595, pp.310-324, Springer-Verlag, 2002.
18. A. Joux, *A one round protocol for tripartite Diffie-Hellman*, ANTS IV, LNCS 1838, pp.385-394, Springer-Verlag, 2000.
19. 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.
20. S. Mitsunari, R. Sakai and M. Kasahara, *A new traitor tracing*, IEICE Trans. Vol. E85-A, No.2, pp.481-484, 2002.
21. D. Pointcheval and J. Stern, *Security arguments for digital signatures and blind signatures*, Journal of Cryptology, Vol.13, No.3, pp.361-396, 2000.
22. R. Sakai, K. Ohgishi and M. Kasahara, *Cryptosystems based on pairing*, SCIS 2000-C20, Jan. 2000. Okinawa, Japan.
23. R. Sakai and M. Kasahara, *Cryptosystems based on pairing over elliptic curve*, SCIS 2003, 8C-1, Jan. 2003. Japan. This paper is available at Cryptology ePrint Archive, http://eprint.iacr.org/2003/054/.
24. 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.
25. E. Verheul, *Self-blindable credential certificates from the Weil pairing*, Advances in Cryptology – Asiacrypt'2001, LNCS 2248, pp. 533–551, Springer-Verlag, 2001.

26. 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.
27. F. Zhang, R. Safavi-Naini and W. Susilo, *An efficient signature scheme from bilinear pairings and its applications*, PKC 2004, Singapore. LNCS, Springer-Verlag, 2004.