An Efficient Verifiably Encrypted Signature Scheme without Random Oracles

Yang Ming[1,2] and Yumin Wang[2]
(Corresponding author: Yang Ming)

School of Information Engineering, Chang'an University[1]
Xi'an, Shaanxi 710064, China (Email: mingyang2001@sohu.com)
State Key Laboratory of Integrated Services Networks, Xidian University[2]
Xi'an, Shaanxi 710071, China

(Received Oct. 09, 2006; revised and accepted July 2, 2007)

Abstract

In this paper, we propose an efficient verifiably encrypted signature scheme based on bilinear pairings. The proposed scheme is proven secure without random oracles. Our scheme has a tight security reduction to a strong but reasonable computational assumption. To the best of our knowledge, it is the third one of this kind in the literature to achieve this security level.

Keywords: Bilinear pairings, random oracles, verifiably encrypted signatures

1 Introduction

A verifiably encrypted signature (VES) provides a way to encrypt a signature under a designated public key and subsequently prove that the resulting ciphertext indeed contains such a signature. It is often used as a building block to construct optimistic fair exchange protocols [2, 11] over Internet, especially online contract signing, e-payment and other electronic commerce. Such a primitive relies on a trusted third party called Adjudicator. In an optimistic way, the adjudication is only needed in cases where a participant attempts to cheat the other or simply crashes. Another key feature of VES is that a participant can always force a fair and timely termination without the cooperation of the participants. Neither party can be left hanging or cheated so long as the adjudicator is available.

When Alice wants to sign a message for Bob but does not want Bob to possess her signature on the message immediately, she 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 stag, Bob can either obtain the signature from Alice or resort to the adjudicator who can reveal Alice's signature.

The concept of VES was introduced by G.Ateniese [5]. In 2003, Boheh et al. [12] and Zhang et al. [10] proposed a verifiably encryption signature with security proofs in the random oracle model based on bilinear pairings, respectively. In 2005, Cheng et al. [7] and Gu et al. [8] presented an ID-based verifiably encrypted signature scheme from bilinear pairings and showed that the scheme was secure in the random oracle model, respectively. In 2006, Zhang et al. [9] pointed out that Gu et al.'scheme [8] was universal forgeable and then proposed a novel verifiably encrypted signature scheme which was proven secure in the random oracle model.

Random oracle model is formal model in analyzing cryptographic schemes, where a hash function is considered as a black-box that contains a random function. However the security in the random oracle model does not imply the security in the real world. Consequently, to design a provable secure verifiably encrypted signature without random oracles is both of theoretical and practical importance. In ICDCIT 2005, M. Choudary Gorantla et al. [3] firstly proposed a verifiably encrypted signature without random oracles. Steve Lu et al. [13] in Eurocrypt 2006 proposed a new verifiably encrypted signature scheme that was provably secure without random oracles.

In this paper, based on Wei's signature [1], we present an efficient verifiably encrypted signature scheme from bilinear pairings which is proven secure without random oracles. Our scheme is more efficient than existing schemes in the literature and then has a tight security reduction.

The rest of the paper is organized as follows: In Section 2, we review some preliminaries. In Section 3, we briefly recall the model and security notions of VES. We propose our VES scheme in Section 4 and provide an analysis about it in Section 5. Finally, we conclude this paper in Section 6.

2 Preliminaries

In this section, we briefly describe bilinear pairings and some related mathematical problems, which form the basis of security for our scheme.

2.1 Bilinear Pairings

Let $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, \cdot)$ be two cyclic groups of order prime q and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a map with the following properties:

1) Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and for all $a, b \in \mathbb{Z}_q$.

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

3) Computability: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

Such a bilinear map is called an admissible bilinear pairing. The Wei pairings and Tate pairings of elliptic curves can be used to construct efficient admissible bilinear pairings.

2.2 The q-Strong Diffie Hellman Assumption

The q-Strong Diffie Hellman (q-SDH) Problem in $\mathbb{G}_1$ [6] is defined as follows: given a (q+1)-tuple $(P, xP, \cdots, x^q P)$ as input, ouput a pair$(c, \frac{1}{x+c}P)$ where $c \in \mathbb{Z}_q^*$. An algorithm $\mathcal{A}$ is said to (t, ε) solve q-SDH problem in $\mathbb{G}_1$ if

$$Pr[\mathcal{A}(P, xP, \cdots, x^q P) = (c, \frac{1}{c+x}P)] \geq \varepsilon.$$

With running time t, where the probability is over the random choice of generator $P \in \mathbb{G}_1$, the random choice of $x \in \mathbb{Z}_q^*$, and the random bits consumed by $\mathcal{A}$.

The (q, t, ε)-SDH assumption says that, no t-time algorithm has advantage at least ε in solving the q-SDH problem in $\mathbb{G}_1$.

2.3 Intractability Assumption about Hash Functions

Let H be a mapping. The H-Collision Problem is to output (m, m') satisfying $m \neq m'$ and $H(m) = H(m')$. An algorithm $\mathcal{A}$ is said to (t, ε)-solve the H-Collision Problem if $\Pr[\mathcal{A}(H) = (m, m') \wedge m \neq m' \wedge H(m) = H(m')] = \varepsilon$ with running time t, and the probability is over random bits $\mathcal{A}$ consumes. H is called a (t, ε)-Collision Resistant hash function if no algorithm can (t, ε)-solve the H-Collision Problem.

3 Verifiably Encrypted Signatures

A verifiably encrypted signature scheme involves three entities: Signer, Verifier, Adjudicator and consists of the following seven algorithms: **KeyGen**, **Sign**, **Verify**, **AdjKenGen**, **VESigGen**, **VESigVerify** and **Adjudication**. The algorithms are described below:

- **KeyGen, Sign, Verify:** As in standard signature schemes.

- **AdjKenGen:** Generate a public-private key pair (APK, ASK) for the adjudicator.

- **VESigGen:** Given a private key SK, a message m and an adjudicator's public key APK, compute a verifiably encrypted signature σ_{VES} on m.

- **VESigVerify:** Given a public key PK, a message m, an adjudicator's public key APK and a verifiably encrypted signature σ_{VES}, verify that σ_{VES} is a valid verifiably encrypted signature on m under public key PK.

- **Adjudication:** Given an adjudicator's key pairs (APK, ASK), a public key PK, and a verifiably encrypted signature σ_{VES} on m, extract and output σ, an ordinary signature on m under public key 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,**VESigGen**(m))

  Verify(m, **Adjudication**(**VESigGen**(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 in polynomial time. The advantage in existentially forging a verifiably encrypted signature of an algorithm $\mathcal{B}$, given access to a verifiably-encrypted-signature generation oracle S and an adjudication oracle A, is

$$AdvVSigF_B \overset{def}{=} \Pr\left[\begin{array}{c} VESigVerify(PK, APK, \\ \qquad m, \sigma_{VES}) = valid \\ (PK, SK) \overset{R}{\leftarrow} KeyGen \\ (APK, ASK) \overset{R}{\leftarrow} AdjKeyGen \\ (m, \sigma_{VES}) \overset{R}{\leftarrow} B^{S,A}(PK, APK) \end{array}\right]$$

  The probability is taken over the coin tosses of the **KeyGen** algorithms, of the oracles, and of the forger. The forger is additionally limited in where its forgery on m must not previously have been queried.

- **Opacity:** This requires that it be difficult to extract an ordinary signature on the same message from verifiably encrypted signature without the help of the Adjudicator. The advantage in extracting a verifiably encrypted signature of an algorithm $\mathcal{F}$, given access to a verifiably-encrypted-signature generation oracle S and an adjudication oracle A, is

$$AdvVSigE_{\mathcal{F}} \overset{def}{=} \Pr\left[\begin{array}{c} Verify(PK, m, \sigma) = valid \\ (PK, SK) \overset{R}{\leftarrow} KeyGen \\ (APK, ASK) \overset{R}{\leftarrow} AdjKeyGen \\ (m, \sigma) \overset{R}{\leftarrow} \mathcal{F}^{S,A}(PK, APK) \end{array}\right]$$

The probability is taken over the coin tosses of the **KeyGen** algorithms, of the oracles, and of the forger. The extraction must be nontrivial: the adversary must not have queried the adjudication oracle A at m.

4 Proposed VES Scheme

In this section, we construct a secure verifiably encrypted signature scheme in the standard model under the q-SDH assumption, which is motivated by Wei's short signature [1].

Let $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, \cdot)$ be two cyclic group of order q, P be a generator of $\mathbb{G}_1$, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear pairing. Let $\{0,1\}^{l_1}$ be the message space, and H be a collision resistant hash $H : \{0,1\}^{l_1} \rightarrow \{0,1\}^{l} \backslash \{0^l\}, l < \log_2 q$.

KeyGen: For a user, pick $x_s, y_s \in \mathbb{Z}_q^*$ at random, and compute $U_s = x_s P, V_s = y_s P, W_s = x_s y_s P$. The user's public key is (U_s, V_s, W_s). The corresponding private key is (x_s, y_s).

Sign: Given a private key (x_s, y_s) of a user and message $m \in \{0,1\}^{l_1}$

1) Randomly generate nonzero $m_1, m_2 \in \{0,1\}^l$ with $m_1 \oplus m_2 = H(m)$.
2) Compute $\sigma = \frac{1}{(x_s+m_1)(y_s+m_2)} P$.

Then, (m_1, σ) is the signature on the message m.

Verify: Given public key (U_s, V_s, W_s), a message $m \in \{0,1\}^{l_1}$, and a signature (m_1, σ). Compute $m_2 = H(m) \oplus m_1$, verify $m_1 \neq 0, m_2 \neq 0$, and $(\sigma, m_2 U_s + m_1 V_s + W_s + m_1 m_2 P) = e(P, P)$.

AdjKenGen: Pick $x_A, y_A \in \mathbb{Z}_q^*$ at random, and compute $U_A = x_A P, V_A = y_A P, W_A = x_A y_A P$. The adjudicator's public key is (U_A, V_A, W_A). The corresponding private key is (x_A, y_A).

VESigGen: Given (x_s, y_s) of signer's private key, a message $m \in \{0,1\}^{l_1}$ and adjudicator's public key (U_A, V_A, W_A).

1) Randomly generate nonzero $m_1, m_2 \in \{0,1\}^l$ with $m_1 \oplus m_2 = H(m)$.
2) Compute $\sigma_{VES} = \frac{1}{(x_s+m_1)(y_s+m_2)} (U_A + V_A + W_A)$.

Then, (m_1, σ_{VES}) is verifiably encrypted signature on the message m.

VESigVerify: Given a verifiably encrypted signature (m_1, σ_{VES}) of a message m, compute $m_2 = H(m) \oplus m_1$, verify $m_1 \neq 0, m_2 \neq 0$, and $e(\sigma_{VES}, m_2 U_s + m_1 V_s + W_s + m_1 m_2 P) = e(U_A + V_A + W_A, P)$.
Adjudication: Given a verifiably encrypted signature(m_1, σ_{VES}) of a message m, the adjudicator can compute $\sigma = \frac{1}{x_A + y_A + x_A y_A} \sigma_{VES}$. Then, (m_1, σ) is the extracted ordinary signature on the message m.

5 Analysis

5.1 Validity

Validity requires that verifiably encrypted signatures verify, and adjudicated verifiably encrypted signatures verify as ordinary signatures.

For a verifiably encrypted signature (m_1, σ_{VES}) on a message m, the validity is easily proven as follows:

$$\begin{aligned} & e(\sigma_{VES}, m_2 U_s + m_1 V_s + W_s + m_1 m_2 P) \\ = \; & e(\frac{1}{(x_s+m_1)(y_s+m_2)}(U_A + V_A + W_A), \\ & \qquad m_2 U_s + m_1 V_s + W_s + m_1 m_2 P) \\ = \; & e(\frac{1}{(x_s+m_1)(y_s+m_2)}(U_A + V_A + W_A), \\ & \qquad (x_s + m_1)(y_s + m_2)P) \\ = \; & e(U_A + V_A + W_A, P). \end{aligned}$$

That is, **VESigVerify**(m,**VESigGen**(m))=1 holds. On the other hand

$$\begin{aligned} & e(\sigma, m_2 U_s + m_1 V_s + W_s + m_1 m_2 P) \\ & = e(\tfrac{1}{(x_s+m_1)(y_s+m_2)} P, m_2 U_s + m_1 V_s + W_s + m_1 m_2 P) \\ & = e(\tfrac{1}{(x_s+m_1)(y_s+m_2)} P, (x_s + m_1)(y_s + m_2)P) \\ & = e(P, P). \end{aligned}$$

Which means **Verify**(m,**Adjudication**(**VESigGen**(m)))=1 holds.

5.2 Security Analysis

In this subsection, we will analyze the security of our proposed scheme and show that the scheme is secure against existential forgery and opaque.

Theorem 1. *Suppose that the Wei's scheme [1] is (t', q_s', ε') secure against existential forgery. Then our proposed verifiably encrypted signature scheme is $(t, q_s, q_A, \varepsilon)$ unforgeable for $t' \leq t + (q_s + q_A + 1)t_{Pm} + (q_A + 1)t_{Inv}$, $q_s' = q_s$ and $\varepsilon' \geq \varepsilon$, where t_{Pm} is the time required to point scalar multiplications in $\mathbb{G}_1$ and t_{Inv} is the time required to inversion in $\mathbb{Z}_q^*$.*

Proof. Suppose that there exists a verifiably encrypted signature forger $\mathcal{B}$, then we can construct a forger algorithm $\mathcal{F}$ for the Wei's scheme.

Setup: $\mathcal{F}$ is given a Wei's signature public key (U_s, V_s, W_s). $\mathcal{F}$ randomly picks $x_A, y_A \in \mathbb{Z}_q^*$, computes $U_A = x_A P, V_A = y_A P, W_A = x_A y_A P$, and provides the adversary $\mathcal{B}$ with (U_s, V_s, W_s) and (U_A, V_A, W_A).

VESigGen Oracle: when $\mathcal{B}$ requests a VES signature on some message m under the adjudicator's public key (U_A, V_A, W_A), $\mathcal{F}$ requests a ordinary signature on m from its own signing oracle and obtains a signature (m_1, σ). Then $\mathcal{F}$ computes $\sigma_{VES} = (x_A + y_A + x_A y_A)\sigma$. The (m_1, σ_{VES}) is a valid verifiably encrypted signature on m. Algorithm $\mathcal{F}$ returns it to $\mathcal{B}$.

Adjudication Oracle: when $\mathcal{B}$ requests adjudication of a verifiably encrypted signature (m_1, σ_{VES}) on some message m under the adjudicator's public key (U_A, V_A, W_A). $\mathcal{F}$ firstly checks that the verifiably encrypted signature is valid, then computes $\sigma = \frac{1}{x_A + y_A + x_A y_A}\sigma_{VES}$ and returns. Note that $\mathcal{F}$ knows the adjudicator's private key (x_A, y_A).

Output: Finally, $\mathcal{B}$ outputs a forged and valid verifiably encrypted signature (m_1^*, σ_{VES}^*) on some message m^* in non-negligible probability. $\mathcal{B}$ must never have made a verifiably encrypted signature generation query at m^*. $\mathcal{F}$ computes $\sigma^* = \frac{1}{x_A + y_A + x_A y_A}\sigma_{VES}^*$ and (m_1^*, σ^*) is therefore a valid Wei's signature on m^*. But, as the Wei's signature scheme is secure against existential forgery under chosen message attack without random oracles (i.e. in the standard model), our proposed verifiably encrypted signature scheme is unforgeable.

Algorithm $\mathcal{F}$ thus succeeds whenever $\mathcal{B}$ does, that is, with probability at least ε. $\mathcal{F}$'s running time is the same as $\mathcal{B}$'s running time plus the time it takes to respond to q_s verifiably encrypted signature generation queries, and q_A adjudication queries, and time to transform $\mathcal{B}$'s final verifiably encrypted signature forgery into a Wei's signature forgery. Each verifiably encrypted signature generation query requires $\mathcal{F}$ to perform one point scalar multiplications in $\mathbb{G}_1$. Each adjudication query requires $\mathcal{F}$ to perform one point scalar multiplications in $\mathbb{G}_1$ and one inversion in $\mathbb{Z}_q^*$. The output phase also requires a point scalar multiplications in $\mathbb{G}_1$ and one inversion in $\mathbb{Z}_q^*$. Hence, the total running time is at most $t' = t + (q_s + q_A + 1)t_{Pm} + (q_A + 1)t_{Inv}$. □

Theorem 2. *The proposed verifiably encrypted signature is $(t, q_s, q_A, \varepsilon)$ opaque if H is $(t + \mathcal{O}(q_s^2), (\varepsilon - q_s q^{-1})/4)$-Collision Resistant hash function and $(q_s, t + \mathcal{O}(q_s^2), (\varepsilon - q_s q^{-1})/4)$-SDH assumption holds.*

Proof. We say that a VES on a message m is opaque if, an algorithm $\mathcal{F}$ cannot extract the ordinary signature (m_1, σ) on the message from a VES (m_1, σ_{VES}). Suppose that $\mathcal{F}$ can extract the Wei's signature (m_1, σ) from a VES (m_1, σ_{VES}), we show how to construct an algorithm $\mathcal{B}$ that solve the intractability problems.

Setup: The algorithm $\mathcal{B}$ is given a random instance $(P, wP, w^2P, \cdots, w^qP)$ of the SDH problem, its goal is to produce a pair $(c, \frac{1}{w+c}P)$ where $c \in \mathbb{Z}_q^*$. $\mathcal{B}$ flips a fair coin $c_{mode} \in \{1, 2\}$ and proceeds below:

1) If $c_{mode} = 1$, $\mathcal{B}$ randomly picks distinct nonzero $\hat{m}_1, \hat{m}_2, \cdots, \hat{m}_{q_s} \in \{0,1\}^l$, and sets $f(w) = \sum_{i=1}^{q_s}(w + \hat{m}_i)$. Expand $f(w)$ and write $f(w) = \sum_{i=0}^{q_s}\alpha_i w^i$ where $\alpha_0, \cdots, \alpha_{q_s}$ are the coefficients of the polynomial $f(w)$. Note that the complexity of the above transformation of the problem instance is $\mathcal{O}(q_s^2)$. $\mathcal{B}$ randomly picks y, y_A, sets $x = w$ and computes $P' = \sum_{i=0}^{q_s}\alpha_i(x^iP) = f(x)P$, $U = \sum_{i=1}^{q_s+1}\alpha_{i-1}(x^iP) = xP'$, $V = yP'$, $W = yU = xyP'$ and adjudicator's public key $U_A = U, V_A = y_AP', W_A = y_AU$.

2) If $c_{mode} = 2$, similarly to $c_{mode} = 1$, $\mathcal{B}$ randomly picks distinct nonzero $\hat{m}_1, \hat{m}_2, \cdots, \hat{m}_{q_s} \in \{0,1\}^l$, and sets $f(w) = \sum_{i=1}^{q_s}(w + \hat{m}_i)$. $\mathcal{B}$ randomly picks x, x_A, sets $y = w$ and computes $P' = \sum_{i=0}^{q_s}\alpha_i(y^iP) = f(y)P$, $U = xP'$, $V = \sum_{i=1}^{q_s+1}\alpha_{i-1}(y^iP) = yP'$, $W = xV = xyP'$ and adjudicator's public key $U_A = x_AP', V_A = V, W_A = x_AV$.

Then $\mathcal{B}$ gives to $\mathcal{F}$ the P', (U, V, W) of signer's public key and adjudicator's public key (U_A, V_A, W_A).

VESigGen Oracle: The $\mathcal{F}$ can issue up to q_s verifiably encrypted signature generation queries in an adaptive fashion. To respond these queries, $\mathcal{B}$ maintains a query counter l which is initially set to 0. Upon receiving a query for m_i, $\mathcal{B}$ increments by one, and checks if $l > q_s$ and $H(m_i) = m_i$. If $l > q_s$ or $H(m_i) = m_i$, it neglects further queries by $\mathcal{F}$ and terminates $\mathcal{F}$. Otherwise,

1) If $c_{mode} = 1$, set $m_{i1} = \hat{m}_i$, $m_{i2} = H(m_i) \oplus m_{i1}$, let $f_i(x) = \frac{f(x)}{(x+\hat{m}_i)} = \sum_{j=1,j\neq i}^{q_s}(x + \hat{m}_j)$. As before, $\mathcal{B}$ expands $f_i(x)$ and writes $f_i(x) = \sum_{j=0}^{q_s-1}\beta_j x^j$, then $\mathcal{B}$ can compute $S_1 = \sum_{j=0}^{q_s-1}\beta_j(x^jP) = f_i(x)P = \frac{1}{x+\hat{m}_i}P'$, $S_2 = \sum_{j=0}^{q_s-1}\beta_j(x^{j+1}P) = xf_i(x)P = \frac{x}{x+\hat{m}_i}P'$. To generate a VES on the message m_i, $\mathcal{B}$ computes $\sigma_{VES_i} = \frac{1}{(y+m_{i2})}S_2 + \frac{y_A}{(y+m_{i2})}S_1 + \frac{y_A}{(y+m_{i2})}S_2$.

2) If $c_{mode} = 2$, set $m_{i2} = \hat{m}_i$, $m_{i1} = H(m_i) \oplus m_{i2}$, let $f_i(y) = \frac{f(y)}{(y+\hat{m}_i)} = \sum_{j=1,j\neq i}^{q_s}(y + \hat{m}_j)$. $\mathcal{B}$ expands $f_i(y)$ and writes $f_i(y) = \sum_{j=0}^{q_s-1}\beta_j y^j$, then $\mathcal{B}$ can compute $S_1 = \sum_{j=0}^{q_s-1}\beta_j(y^jP) = f_i(y)P = \frac{1}{y+\hat{m}_i}P'$, $S_2 = \sum_{j=0}^{q_s-1}\beta_j(y^{j+1}P) = yf_i(y)P = \frac{y}{y+\hat{m}_i}P'$. To generate a VES on the message m_i, $\mathcal{B}$ computes $\sigma_{VES_i} = \frac{x_A}{(x+m_{i1})}S_1 + \frac{1}{(x+m_{i1})}S_2 + \frac{x_A}{(x+m_{i1})}S_2$.

In either case, $\mathcal{B}$ returns to $\mathcal{F}$ the verifiably encrypted signature (m_{1i}, σ_{VES_i}).

Table 1: Comparison of Seven VES Schemes

Scheme	Size	VESignGen	VESignVerify	Adjudication	R.O.
[12]	320bit	$2M$	$3e$	$1M$	YES
[10]	160bit	$1M$	$1e$+$1M$	$1M$	YES
[7]	480bit	$2M$	$3e$+$1m$	$1M$	YES
[9]	480bit	$1e$+$1E$+$2M$	$4e$+$1E$+$2M$	$1e$+$1E$+$1M$	YES
[3]	320bit	$2M$	$2e$	$1M$	NO
[13]	480bit	$3M$	$3e$	$1M$	NO
ours	320bit	$3M$	$1e$	$1M$	NO

Adjudication Oracle: when $\mathcal{F}$ requests adjudication of a verifiably encrypted signature (m_{1i}, σ_{VES_i}) on the message m_i under the adjudicator's public key (U_A, V_A, W_A). $\mathcal{B}$ first verifies that (m_{1i}, σ_{VES_i}) is valid and rejects it otherwise.

1) If $c_{mode} = 1$, then $m_{i1} = \hat{m}_i$. $\mathcal{B}$ can compute $m_{i2} = H(m_i) \oplus m_{i1}$, $\sigma_i = \frac{1}{(y+m_{i2})} S_1$.

2) If $c_{mode} = 2$, then $m_{i2} = \hat{m}_i$. $\mathcal{B}$ can compute $m_{i1} = H(m_i) \oplus m_{i2}$, $\sigma_i = \frac{1}{(x+m_{i1})} S_1$.

In either case, $\mathcal{B}$ returns to $\mathcal{F}$ the ordinary signature (m_{i1}, σ_i).

Output: Finally, $\mathcal{F}$ outputs a valid ordinary message signature pair (m_1^*, σ^*) on the message m^*, $m^* \neq m_i$, $1 \leq i \leq q_s$. Compute $m_2^* = H(m^*) \oplus m_1^*$. When the pair is valid, one of the following events must happen:

Event A1: $m_1^* \neq \hat{m}_i$ for any i. If $c_{mode} = 1$, then we have $(\sigma^*, (x+m_1^*)(y+m_2^*)P') = ((y+m_2^*)\sigma^*, (x+m_1^*)P') = e(P', P')$. And $(y+m_2^*)\sigma^* = \frac{1}{x+m_1^*} P' = \frac{f(x)}{x+m_1^*} P$ thus $(m_1^*, f^{-1}(x)(y+m_2^*)\sigma^*)$ is a solution of SDH Problem.

Event A2: $m_2^* \neq \hat{m}_i$ for any i. If $c_{mode} = 2$, then we have $(\sigma^*, (x+m_1^*)(y+m_2^*)P') = ((x+m_1^*)\sigma^*, (y+m_2^*)P') = e(P', P')$. And $(x+m_1^*)\sigma^* = \frac{1}{y+m_2^*} P' = \frac{f(y)}{y+m_2^*} P$ thus $(m_2^*, f^{-1}(y)(x+m_1^*)\sigma^*)$ is a solution of SDH Problem.

Event B: $m_1^* = \hat{m}_i, m_2^* = \hat{m}_{i'}$ for some $1 \leq i, i' \leq q_s, i = i'$. Then $H(m^*) = H(m_i)$ and the tuple (m^*, m_i) solves the H-Collision Problem.

This completes the description of algorithm $\mathcal{B}$. A standard argument shows that if $\mathcal{B}$ does not abort, then, from the viewpoint of $\mathcal{F}$, the simulation provided by $\mathcal{B}$ is indistinguishable from a real attack scenario. Therefore, $\mathcal{F}$ produces a valid forgery in time t with probability at least ε. The probability of each event is independent of the value of c_{mode}, due to the negligibility of the simulation deviation. The sum of the probabilities of all events above is greater than or equal to ε. Then, at least one of the following composite event has probability lower bounded by $\frac{\varepsilon}{4} - \frac{q_s}{q}$

1) $\{\{$**EventA1** $\wedge\, c_{\bmod e} = 1\} \vee \{$**EventA2** $\wedge\, c_{\bmod e} = 2\}\} \wedge \mathcal{F}$ forges.

2) **EventB** $\wedge\, \mathcal{F}$ forges.

Note that the total probability of aborting during **VESigGen Oracle** simulation is $\frac{q_s}{q}$. The theorem is then obtained. □

5.3 Efficiency

In this section, we compare our proposed scheme with previous schemes in the literature. For the comparison, we instantiate pairing-based schemes using Barreto-Naehring curves [4] with 160-bit point representation. The **Size** column gives verifiably encrypted signature length at the 1024-bit security level. The **VESignGen**, **VESigVerify** and **Adjudication** columns give the computational costs of those operations. " R.O." denotes whether the security proof uses random oracles. Denote M a scalar multiplication in $\mathbb{G}_1$, E an Exponent operation in $\mathbb{G}_2$, and e a computation of the pairing. We do not take other operations into account. The computation overheads of our VES and the schemes in [3, 7, 9, 10, 12, 13] (optimized by precomputing) are summarized in Table 1.

We note that the computation of the pairing is the most-consuming. Although there have been many papers discussing the complexity of pairings and how to speed up the pairing computation, the pairing computation is the operation which by far takes the most running time. In our scheme, we can precompute $e(U_A + V_A + W_A, P)$ and publish it as party of the adjudicator's public keys. Therefore, there is only one pairing computation in **VESigVerify** phase, but there are two and three pairings computation in [3] and [13] respectively. Finally, our VES scheme is shorter than [13] and as long as [3]. In three existing VES schemes without random oracles, our scheme is most efficient.

6 Conclusion

Verifiably encrypted signatures are very important cryptographic primitives, and are used in optimistic contract signing protocols to enable fair exchange. In this paper, we proposed an efficient verifiably encrypted signature which is proven to be secure without random oracles. Our scheme is more efficient than existing schemes in the literature.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Nature Science Foundation of China (No. 60473027).

References

[1] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures," *IEEE Selected Areas in Comm.*, vol. 18, no. 4, pp. 593-610, 2000.

[2] G. Ateniese, "Verifiable encryption of digital signature and applications," *ACM Transactions on Information and System Security*, vol. l7, no. 1, pp. 1-20, 2004.

[3] F. Bao, R. Deng, and W. Mao, "Efficient and practical fair exchange protocols with offline TTP," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 77-85, 1998.

[4] P. Barreto, and M. Naehrig, "Pairing-friendly elliptic curves of prime order," *SAC'05*, LNCS 3897, pp. 319-331, Springer-Verlag, 2006.

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," *Eurocrypt'03*, LNCS 2656, pp. 416-432, Springer-Verlag, 2003.

[6] D. Boneh, and X. Boyen, "Short signatures without random oracles," *Eurocrypt'04*, LNCS 3027, pp. 56-73, Springer-Verlag, 2004.

[7] X. Cheng, J. Liu, and X. Wang, "Identity-based aggregate and verifiably encrypted signatures from bilinear pairing," *ICCSA '05*, LNCS 3483, pp. 1046-1054, Springer-Verlag, 2005.

[8] M. C. Gorantla, and A. Saxena, "Verifiably encrypted signature scheme without random oracles," *ICDCIT '05*, LNCS 3816, pp. 357-363, Springer-Verlag, 2005.

[9] C. Gu, and Y. Zhu, "An ID-based verifiable encrypted signature scheme based on Hess's scheme," *CISC'05*, LNCS 3822, pp. 42-52, Springer-Verlag, 2005.

[10] S. Lu, R. Ostrovsky, and A. Sahai, et al., "Sequential aggregate signatures and multisignatures without random oracles," *Eurocrypt '06*, LNCS 4004, pp. 465-485, Springer-Verlag, 2006.

[11] V. K. Wei, and T. H. Yuen, "More short signatures without random oracles," *Cryptology ePrint Archive, Report 2005*, vol. 463, 2005. (Accepted by International Journal of Network Security) (http://eprint.iacr.org/2005/463.)

[12] F. Zhang, R. Safavi-Naini, and W. Susilo, "Efficient verifiably encrypted signature and partially blind signature from bilinear pairings," *Indocrypt'03*, LNCS 2904, pp. 191-204, Springer-Verlag, 2003.

[13] J. Zhang, and W. Zou, "A robust verifiably encrypted signature scheme," *EUC Workshops '06*, LNCS 4097, pp. 731-740, Springer-Verlag, 2006.

Yang Ming received his B.S. and M.S. degrees at the Department of Science from Xi'an University of Technology, China, in 2002 and 2005 respectively. He received the Ph.D. degree in cryptography from Xidian University in 2008. He is currently a lecturer in School of Information Engineering, Chang'an University. His research interests include cryptography and network security.

Yumin Wang is now a professor and PhD supervisor at the Department of Telecommunications Engineering of Xidian University. His main research areas are information theory, coding, and cryptography etc..