# A Robust Verifiably Encrypted Signature Scheme

Jianhong Zhang[1,2] and Wei Zou[1]

[1] Institute of Computer Science & Technology, Peking University, Beijing 100871, P.R. China
{zhangjianhong, zouwei}@icst.pku.edu.cn
[2] Institute of Image Processing and Pattern Recognition, North China University of Technology, Beijing 100041, P.R. China

**Abstract.** A verifiably encrypted signature can convince the verifier that a given cipher-text is the encryption of a signature on a given message. It is often used as a building block to construct optimistic fair exchange. Recently, Gu *et.al* gave an ID-based verifiably encrypted signature scheme and claimed that their scheme was secure in random oracle model. Unfortunately, in this works, we show that their scheme is insecure. And we can mount to universal forgery attack in their model. In other words, any one is able to forge a verifiably encrypted signature on arbitrary message $m$. Subsequently, a novel verifiably encrypted signature scheme (VES) is proposed and the scheme is proven secure in random oracle model. Moreover, the size of verifiably encrypted signature in our scheme is shorter than that of Gu *et.al*'s signature.

## 1 Introduction

A verifiably encrypted signature, which was proposed by N.Asokan [7], 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 enables to construct optimistic fair exchange [4,5] over the Internet, and relies on a trusted third party call Adjudicator, in optimistic way, that 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 other participants. Neither party can be left hanging or cheated so long as the adjudicator is available.

A valid VES can convince the verifier that a given cipher-text is the encryption of a signature on a given message. Alice creates a VES on a message by using her private key and an Adjudicator's public key. Bob is convinced that the encrypted signature is indeed of Alice, which he verifies using the public key of Alice and the Adjudicator. Even though Bob does not have the capability of decrypting the VES, the verification is performed without revealing any information about Alice's signature. If a dispute, the adjudicator can extract Alice's signature from VES on the message.

Since the concept of VES was included,J.Camenish[6] and G.Ateniese [7] proposed a verifiable encryption signature based on the discrete logarithm problem,

X. Zhou et al. (Eds.): EUC Workshops 2006, LNCS 4097, pp. 731–740, 2006.

respectively. In 2003, Boheh *et.al* [8] and Zhang *et.al*[9] proposed a verifiably encryption signature with security proofs in random oracle based on bilinear Pairings, respectively. In ICDCIT 2005, M.Choudary Gorantla *et.al*[2] proposed a novel verifiably encrypted signature without random oracle. Recently, by combining ID-based public key cryptography with verifiably encrypted signature, Gu *et.al*[1] proposed a ID-based verifiably encrypted signature scheme based on Hess' signature scheme (Gu *et.al* scheme for short), and claimed that their scheme was secure in random oracle model. Unfortunately, in this work, we show that Gu et.al scheme is universal forgeable. Namely, any one can forge a verifiably encrypted signature on arbitrary a message. Subsequently, we propose a novel secure verifiably encrypted signature scheme, and the scheme is proven secure in random oracle model. Moreover, the size of verifiably encrypted signature in our scheme is shorter than that of Gu *et.al*'s signature.

The rest of the paper is organized as follows. In section 2, we review some preliminaries . In section 3, we briefly recall Gu *et.al* scheme and show that the scheme is insecure in section 4. In section 5, we propose our VES scheme and prove it to be secure in section 6. Finally, we draw this paper.

## 2 Preliminaries

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

Let $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of order $q$. Let $e : G_1 \times G_1 \rightarrow G_2$ be a map which the following properties.

- Bilinear: $\forall P, Q \in G_1$, and $a, b \in Z_q$, $e(aP, bP) = e(P, P)^{ab}$
- Non-degeneracy: There exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$,in other words, the map doesn't send all pair in $G_1 \times G_1$ to the identity in $G_2$.
- Computable: There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in G_1$.

Such a bilinear map is called an admissible bilinear pairing. The Weil Parings and Tate Pairings of elliptic curves are able to construct an efficient admissible pairings.

The security of our proposed scheme is related to the computational Diffie-Hellman problem (CDHP), which is given below.

**Definition 1.** *(Computational Diffie-Hellman Problem) Let $a, b$ be chosen from $Z_q$ at random and $P$ be chosen from $G_1$ at random. Given $(P, aP, bP)$, compute $abP \in G_1$.*

The success probability of any probabilistic polynomial-time $\mathcal{A}$ in solving CDH problem in $G_1$ is defined to be

$$Succ^{CDH}_{\mathcal{A},G_1} = Pr[\mathcal{A}(P, aP, bP) = abP : a, b \in Z_p]$$

The CDH assumption states that for every probabilistic polynomial-time algorithm $\mathcal{A}$, $Succ^{CDH}_{\mathcal{A},G_1}$ is negligible.

**Definition 2.** *A verifiably encrypted signature scheme consists of the following algorithms: KeyGen, Sign, Verify, AdjKenGen, VESig Generate, VESig Verify and Adjudicate. The algorithms are described below.*

- **Key Generation, Signing, Verification.** As in standard signature schemes.
- **Adjudicator Key.** Generate a public-private key pair $(Q_A, s_A)$ for the adjudicator.
- **VESig Generation.** Give a private key $D$, a message $M$, and an adjudicator's $Q_A$, compute a verifiably encrypted signature $\delta$ on $M$.
- **VESig Verification.** Given a public key $Q$, a message $M$, an adjudicator's public key $Q_A$ and a verifiably encrypted signature $\delta$, verify that $\delta$ is a valid verifiably encrypted signature on $M$ under public key $Q$.
- **Adjudication.** Given an adjudicator's key pairs $(Q_A, s_A)$, a public key $Q$, and a verifiably encrypted signature $\delta$ on some message $M$, extract and output $\delta_1$, an ordinary signature on $M$ under public key $Q$.

A verifiably encrypted signature scheme should satisfy the following properties;

- Unfogeability: It is difficult to forge a valid verifiably encrypted signature in polynomial time.The advantage in existentially forging a verifiably encrypted signature of an algorithm $F$, given access to a verifiably-encrypted-signature creation oracle $S$ and an adjudication oracle $A$, along with a hash oracle, is

$$AdvVSF_f \stackrel{\text{def}}{=} Pr\left[\begin{array}{c} VESigVerify(PK, APK, M, w) = valid \\ (PK, SK) \stackrel{\text{R}}{\leftarrow} KeyGen, \\ (APK, ASK) \stackrel{\text{R}}{\leftarrow} AdjKeyGen, \\ (M, w) \stackrel{\text{R}}{\leftarrow} F_{S,A}(PK, APK) \end{array}\right]$$

  The probability is taken over the coin tosses of the key-generation 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: Given a verifiably encrypted signature, it is difficult to extract the ordinary signature on the same message from the verifiably encrypted signature time without the help of the **Adjudicator**. The advantage in extracting a verifiably encrypted signature of an algorithm $\varepsilon$ , given access to a verifiably-encrypted-signature creation oracle $S$ and an adjudication oracle $A$, along with a hash oracle, is

$$AdvVSF_\varepsilon \stackrel{\text{def}}{=} Pr\left[\begin{array}{c} Verify(PK, M, \sigma) = valid \\ (PK, SK) \stackrel{\text{R}}{\leftarrow} KeyGen, \\ (APK, ASK) \stackrel{\text{R}}{\leftarrow} AdjKeyGen, \\ (M, \sigma) \stackrel{\text{R}}{\leftarrow} \varepsilon_{S,A}(PK, APK) \end{array}\right]$$

  The probability is taken over the coin tosses of the key-generation algorithms, of the oracles, and of the forger. The extraction must be nontrivial: the adversary must not have queried the adjudication oracle $A$ at $M$.

## 3 Review of Gu *et.al* VES Scheme

In the section, we briefly recall *Gu et.al*'s scheme. Please the interested reader refer to [1] for general reference. Private keys are generated for entities by a trusted third party called a private key generator (**PKG** for short ).

Let $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of order $q$, $P$ be a generator of $G_1$, $e : G_1 \times G_1 \longrightarrow G_2$ be an admissible bilinear pairing. Gu-Zhu scheme consists of the following algorithms:

- **Setup:** Given $(G_1, G_2, q, e, P)$, randomly choose a number $s \in Z_q^*$ and set $P_{pub} = sP$. Then choose three hash functions $H_1 : \{0,1\}^* \longrightarrow G_1^*$, $H_2 : \{0,1\}^* \longrightarrow Z_q^*$ and $H_3 : G_2 \longrightarrow Z_q$. The system parameters $\Omega = (G_1, G_2, q, e, P, P_{pub}, H_1, H_2, H_3)$. The master key is $s$.
- **Extract:** Given an identity $ID_X \in \{0,1\}^*$, compute $Q_X = H_1(ID_X) \in G_1$,$D_X = sQ_X$. PKG uses this algorithm to extract the user's private key $D_X$, and gives $D_X$ to the user by a secure channel.
- **Sign:** Given a private key $D_X$ of a user and a message $m$, pick $k \in Z_q^*$ at random, and output a signature $(r, U)$, where $r = e(P,P)^k$, $h = H_2(m, r)$ and $U = hD_X + kP$.
- **Verify:** Given a signature $(r, U)$ of an identity $ID_X$ on a message $m$, compute $h = H_2(m, r)$ and check whether the following equation holds

$$r = e(U, P) \cdot e(H_1(ID_X), P_{pub})^{-h}$$

- **VE-Sign:** Given a secret key $D_X$, a message $m \in \{0,1\}^*$ and an adjudicator's identity $ID_A$.
  1. choose $k_1, k_2 \in Z_q^*$ at random.
  2. compute $r = e(P,P)^{k_1}$, $h = H_2(m, r), h' = H_3(e(Q_A, P_{pub})^{k_2})$
  3. compute $U_1 = h'P, U_2 = k_2 P, V = hD_X + (k_1 + h'k_2)P + h'Q_A$
  4. output the verifiably encrypted signature $(r, V, U_1, U_2)$.
- **VE-Verify:** Given a verifiably encrypted signature $(r, V, U_1, U_2)$ of a message $m$, compute $h = H_2(m, r)$, and accept this signature if and only if

$$e(P, V) = r \cdot e(hP_{pub}, Q_X) \cdot e(U_1, Q_A + U_2) \tag{1}$$

- **Adjudication:** Given the adjudicator's secret key $D_A$, and a valid verifiably encrypted signature $(r, V, U_1, U_2)$ on the message $m$, compute $U = V - H_3(e(D_A, U_2))(Q_A + U_2)$, finally, output the original signature $(r, U)$.

## 4 Security Analysis

Gu *et.al* claimed that their scheme[1] was secure in random oracle model. In this section, we will show the scheme is universally forgeable by analyzing its security. Note that any one can forge a verifiable encryption signature on arbitrary a message without the private key $D_X$ of the user with the identity $ID_X$. To forge a verifiably encryption signature of the user with the identity $ID_X$, The forging procedures are as follows:

1. Given arbitrary a message $m$, randomly choose $r_2 \in Z_q$, then compute $r = e(P, r_2 Q_X)$,$h = H_2(m, r)$ and $Q_X = H_1(ID_X)$ .
2. randomly choose $r_3 \in Z_q$ and set $U_2 = -Q_A + r_3 Q_X$.
3. randomly choose $r_4 \in Z_q$ and compute $U_1 = \frac{-(hP_{pub} + r_4 P) + P}{r_3}$
4. set $V = (1 + r_2 - r_4)Q_X$

Then the forged verifiably encryption signature is $(r, V, U_1, U_2)$.

If the forged signature $(r, V, U_1, U_2)$ satisfies the verification equation (1) of the above verifiably encryption signature, then it means that the forged signature is valid. In the following, we show that the above forgery attack satisfies the verification equation (1). Since

$$\begin{aligned} r \cdot e(hP_{pub}, Q_X) \cdot e(U_1, Q_A + U_2) &= r \cdot e(hP_{pub}, Q_X) \cdot e(U_1, Q_A + (-Q_A + r_3 Q_X)) \\ &= r \cdot e(hP_{pub}, Q_X) \cdot e(U_1, r_3 Q_X) \\ &= r \cdot e(hP_{pub}, Q_X) \cdot e(\frac{-(hP_{pub} + r_4 P) + P}{r_3}, r_3 Q_X) \\ &= r \cdot e(hP_{pub}, Q_X) \cdot e(-(hP_{pub} + r_4 P) + P, Q_X) \\ &= e(P, (1 + r_2 - r_4)Q_X) \\ &= e(P, V) \end{aligned}$$

Thus, our forged verifiably encryption signature can satisfy the verifying phase of Gu *et.al*'s scheme. It implies that our forgery attack is successful. An important reason to such attack is that any one that possesses of the user's original signature $(U, r)$ on the message $m$, can produce a verifiably encrypted signature on the message $m$. Thus, if the Hess's signature is forgeable, then we can forge a verifiably encrypted signature; but if Gu et.al scheme is forgeable, then it don't mean the Hess's signature scheme is forgeable. Therefore, the Hess's signature scheme is not equivalent to Gu et.al scheme.

According to the verification, we can know that Gu *et.al*'s scheme is insecure, any one can forge a verifiably encryption signature on arbitrary a message $m$. The reason to mount to the attack is that the relation between $U_1$ and $U_2$ cannot be reflected in the verification phase of verifiably encryption signature. $U_1$ and $U_2$ is mutually independent in the verifying equation.

## 5 Our Proposed Scheme

In the section, we will propose a novel ID-based verifiably encryption signature scheme. We assume that $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of order $q$, $P$ be a generator of $G_1$, $e : G_1 \times G_1 \longrightarrow G_2$ be an admissible bilinear pairing. The detail procedure is as follows:

**Setup:** Given $(G_1, G_2, q, e, P)$, randomly choose a number $s \in Z_q^*$ and set $P_{pub} = sP$. The adjudicator randomly chooses a $s_A$ as his private key and compute his public key $Q_A = s_A P$ . Then choose three hash functions $H_1 : \{0,1\}^* \longrightarrow G_1^*$, $H_2 : \{0,1\}^* \longrightarrow Z_q^*$ and $H_3 : G_1 \times G_2 \longrightarrow Z_q$. The system parameters $\Omega = (G_1, G_2, q, e, P, P_{pub}, H_1, H_2, H_3)$. The master key is $s$.

**Extract:** Given an identity $ID_X \in \{0,1\}^*$ of a user, compute $Q_X = H_1(ID_X) \in G_1$,$D_X = sQ_X$. PKG uses this algorithm to extract the user's private key $D_X$, and gives $D_X$ to the user by a secure channel.
**Sign phase:** Given a private key $D_X$ of a user and message $m$, pick $k \in Z_q^*$ at random, and compute

$$\begin{aligned} r_1 &= e(P, kP) \\ h &= H_2(m, r_1) \\ U &= hD_X + kP \end{aligned}$$

then, $(r_1, U)$ is the signature on the message $m$.
**Verify phase:** Given a signature $(r_1, U)$ of an identity $ID_X$ for a message $m$, compute $h = H_2(m, r)$ and accept the signature if and only if

$$r = e(U, P) \cdot e(H_1(ID_X), P_{pub})^{-h}$$

**VE-Sign:** Given a secret key $D_X$, a message $m \in \{0,1\}^*$ and an adjudicator's public key $Q_A$. He computes as follows:

- choose two random numbers $k_1, k_2 \in Z_q$.
- compute $r = e(P, P)^{k_1}$, $h = H_2(m, r \cdot e(k_2 P, P))$
- compute $U_2 = k_2 P, V = hD_X + k_1 P + k_2 H_3(U_2, r) Q_A$.
- output the verifiably encrypted signature $(r, V, U_2)$ on the message $m$.

**VE-Verify:** Given a verifiable encrypted signature $(r, V, U_2)$ of a message $m$, a verifier first computes $h = H_2(m, r \cdot e(U_2, P))$, and accepts the signature if and only if

$$e(P, V) = r \cdot e(hP_{pub}, Q_X) \cdot e(U_2, H_3(U_2, r) Q_A)$$

**Adjudication:** Given a verifiably encrypted signature $(r, U_2, V)$ on a message $m$, the adjudicator can extract as follows: he computes $U = V - s_A H_3(U_2, r) U_2 + U_2 = V - (s_A H_3(U_2, r) - 1) U_2$, $r_1 = r \cdot e(U_2, P)$. Then $(U, r_1)$ is the extracted signature on the message $m$.

## 6 Analysis

In this section, we first justify the validity of the scheme and subsequently the security of the scheme.
**[Correction]**
It is obvious, the detail analysis is given in full paper.
**[Security Analysis]**
In the subsection, we will analyze the security of our proposed scheme and show the scheme is secure against existential forgery and extraction .

**Theorem 1.** *Let $G_1$ and $G_2$ be cyclic groups of prime order $p$ with a computable bilinear map $e : G_1 \times G_1 \longrightarrow G_2$, $P$ be a generator of group $G_1$. Suppose that the Hess's signature scheme[13] is $(t', q'_H, q'_S, \epsilon')-$ secure against*

*existential forgery. Then the our proposed verifiably encrypted signature scheme is* $(t, q_{H_1}, q_{H_2}, q_{H_3}, q_{VS}, q_A, \epsilon)-$ *secure against existential forgery, where* $t \leq t' - ((3q_{VS} + q_A + 1)c_{G_1} + (q_{VS} + q_A + 1)c_{G_2})$.

*Proof.* Suppose that there exists a verifiably-encrypted-signature forger algorithm $\mathcal{V}$, then we can construct a forger algorithm $\mathcal{F}$ for the Hess signature scheme. (**Note that**: the Hess signature is proven secure in random oracle model, please interested reader refer [11]to the complete content).

The Hess-signature forger $\mathcal{F}$ is given a public key $ID_X$, and has access to a signing oracle for $ID_X$ and two hash oracles. It simulates the challenger and runs interacts with $\mathcal{V}$ as follows:

- Setup: A challenger $\mathcal{C}$ runs **Setup** of Hess signature scheme, and gives the corresponding system parameters $(G_1, G_2, q, e, P, P_{pub}, H_1, H_2)$ to $\mathcal{F}$. And algorithm $\mathcal{F}$ chooses a $s_A \in Z_q$ at random and computes $P_A = s_A P$. Let $(s_A, P_A)$ serve as the adjudicator's key pairs. Select a hash function $H_3 : G_1 \times G_2 \longrightarrow Z_q$. Let $(G_1, G_2, q, e, P, P_{pub}, H_1, H_2, H_3, (s_A, P_A))$ be input of $\mathcal{V}$.
- $H_1(\cdot), H_2(\cdot), E(\cdot) - Oracle$ : When algorithm $\mathcal{V}$ request a $H_1(\cdot)$,(or $H_2(\cdot)$, $E(\cdot)$) query, Algorithm $\mathcal{F}$ makes a query to its own hash oracle $H_1(\cdot)$,(or $H_2(\cdot), E(\cdot)$), receiving some value, with which it responds to $\mathcal{V}$'s query.
- $H_3-$ Oracle: for a query $(U_{2_i}, r_i)$ to $H_3$, $\mathcal{F}$ first checks whether $H_3(U_{2_i}, r_i)$ is defined. if it exists, then return the corresponding value. If not, randomly choose $h'_i \in Z_q$, and set $H_3(U_{2_i}, r_i)) = h'_i$ and return $h'_i$ to $\mathcal{V}$.
- **VESig Oracle:** When algorithm $\mathcal{V}$ requests a signature on some string $M$ under the user with the identity $ID_X$ and the adjudicator' public $P_A$, algorithm $\mathcal{F}$ queries its signing oracle with input $(ID_X, M)$ and obtains the reply $(U, r)$. Then $\mathcal{F}$ computes as follows:
  1. randomly choose $k_1 \in Z_q$.
  2. compute $r' = r \cdot e(P, -k_1 P)$ and $U'_2 = k_1 P$.
  3. then compute $V' = U - k_1 P + k_1 H_3(U'_2, r')Q_A$

  Finally, the algorithm $\mathcal{F}$ returns $(r', U'_2, V')$ to VESignature on the string $M$.
- **Adjudication Oracle:** Algorithm $\mathcal{V}$ requests adjudication for $(r, U_2, V)$, a verifiably encrypted signature on a message $M$ under the user with the identity $ID_X$ and adjudicator key $P_A$. Algorithm $\mathcal{F}$ checks that the verifiably encrypted signature is valid, then computes as follows:

$$U = V - s_A \cdot H(U_2, r)U_2 + U_2, r_1 = r \cdot e(U_2, P) \tag{2}$$

- **Output:** Finally, if $\mathcal{V}$ outputs a valid and nontrivial verifiably encrypted signature $(r_1^*, U^*, V^*)$ on a message $M^*$ in non-negligible probability. $\mathcal{F}$ sets $U^* = V^* - s_A \cdot H(U_2^*, r^*)U_2^* + U_2^*, r_1^* = r^* \cdot e(U_2^*, P)$, which is a valid Hess signature on the message $M^*$.

It remains only to analyze the success probability and running time of $\mathcal{F}$. Algorithm $\mathcal{F}$ succeeds when $\mathcal{V}$ does, that is, with probability at least $\epsilon$.

Algorithm $\mathcal{F}$'s running time is the same as $V$'s running time plus the time it takes to respond to $q_{H_1}, q_{H_2}, q_{H_3}$ hash queries, $q_{VS}$ verifiably-encrypted signature queries, and $q_A$ adjudication queries, and the time to transform $V$'s final verifiably-encrypted signature forgery into a Hess signature forgery. Hash queries impose no overhead. Each verifiably-encrypted signature query requires $\mathcal{F}$ to perform three point multiplications in $G_1$ and an exponentiation in $G_2$. Each adjudication query requires $\mathcal{F}$ to perform an point multiplication in $G_1$ and an exponentiation in $G_2$. The output phase also requires an exponentiation and an point multiplication. We assume that exponentiation in $G_2$ takes time $c_{G_2}$, and point multiplication in $G_1$ takes time $c_{G_1}$. Hence, the total running time is at most $t + (3q_{VS} + q_A + 1)c_{G_1} + (q_{VS} + q_A + 1)c_{G_2}$. □

**Theorem 2.** *Let $G_1$ and $G_2$ be cyclic groups of prime order $p$ with a computable bilinear map $e : G_1 \times G_1 \longrightarrow G_2$, $P$ be a generator of group $G_1$. $Q_A$ is the adjudicator's public key. Suppose that the CDH problem isn't solved in a polynomial time, then our verifiably encrypted signature is secure against extraction.*

*Proof.* We say that a VES (Verifiably Encrypted Signature) on a message $M$ is secure against extraction, if an adversary $\mathcal{A}$ cannot extract the original signature $(U, r_1)$ on the message from a VES $(r, U_2, V)$. Let us recall the CDH problem as follows. Given a randomly chosen $P \in G_1$, as well as $aP, bP$ (for unknown randomly chosen $a, b \in Z_q$), compute $abP$. To show the proof, we suppose that there is a polynomial algorithm $\mathcal{A}$ that can extract the original signature $(U, r_1)$ from a VES $(r, U_2, V)$. The algorithm $\mathcal{A}$ is given the inputs: an **ID** of the user, $ID_X$, a message $M$, a public key $P_{Pub}$ of the system and an adjudicator's public key $Q_A$. Finally, it outputs a valid the original signature $(U, r_1)$, where

$$Pr[Verification(m, (U, r_1), ID_X, P_{Pub}) = accept] = 1$$

In the following, we show how to use this algorithm to solve the CDH problem.

In our setting, we know the public information $P_{Pub}, ID_X, Q_A$. From this public information, we can obtain $Q_X = H_1(ID_X)$. Since $P$ is a generator in $G_1$, then we can rewrite the following parameters as

$$Q_X = aP, P_{Pub} = bP$$

Now, we construct an algorithm $\mathcal{F}$ to solve the CDH problem. Algorithm $\mathcal{F}$ will control $\mathcal{A}$ and replace $\mathcal{A}$'s interaction with the user (the signer) by simulation. Firstly, $\mathcal{F}$ generates a list of **ID** of its choice, together with a random $s_i$ associated with it. The idea of the game is illustrated as follows. The purpose of $\mathcal{F}$ is to inject the information above $(aP, bP)$ during the simulation. Without loss of generality, we only show the interaction where $\mathcal{A}$ interacts with $\mathcal{F}$ for the information that $\mathcal{F}$ wants. There is a probability that $\mathcal{F}$ will fail, i.e. when $\mathcal{A}$ queries the private key for $ID_X$, which will match with the published $P_{Pub}$, and when $\mathcal{A}$ queries **Adjudicator Oracle** by a VES under the user with identity $ID_X$ and the adjudicator's public key $Q_A$. Since $\mathcal{F}$ hasn't this information, then $\mathcal{F}$ will halt the game. $\mathcal{A}$ will be run twice with a different random query set, but from the same list **ID**'s generated at the first place. The attack is successful, when $\mathcal{A}$

outputs two signatures from the given parameters by forking Lemma[10]. The game is described as follows:

**$H_1$-Hash Oracle:** When $\mathcal{A}$ queries $H_1(\cdot)$ oracle with the input $ID_i$, $\mathcal{F}$ checks whether $H_1(ID_i)$ is defined or not. If not, he defines $H_1(ID_i) = \begin{cases} aP, & \text{i=X} \\ s_iP, & \text{i} \neq \text{X} \end{cases}$ and returns $H_1(ID_i)$ to $\mathcal{A}$.
**$H_2$−Hash Oracle:** When querying $(m_i, r_i)$ to $H_2(\cdot)$, if $h_i = H_2(m_i, r_i)$ exists, then $\mathcal{F}$ returns $h_i$; otherwise, randomly chooses $h_i \in Z_q$ to return it.
**$H_3$−Hash Oracle:** When querying $(U_{2_i}, \tilde{r}_i)$ to $H_3(\cdot)$, if $h_{3_i} = H_3(U_{i_2}, \tilde{r}_i)$ exists, then $\mathcal{F}$ returns $h_{3_i}$; otherwise, randomly chooses $h_{3_i} \in Z_q$ to return it.
*Extract*−**Oracle:** When querying $ID_i$ to Extract oracle, if $i = X$, $\mathcal{F}$ halts it. Otherwise, $\mathcal{F}$ sets $D_i = s_iP_{Pub}$ as the reply to $\mathcal{A}$.
**VSignature Oracle:** For input a user's identity $ID_i$, an adjudicator's public key $Q_A$ and a message $m_i$, $\mathcal{F}$ responses as follows:

- if $i = X$, then $\mathcal{F}$
  1. randomly chooses $l_i, h_i, k_{2_i} \in Z_q$ to compute $r_i = e(l_iP, P)e(h_iP_{Pub}, Q_X)^{-1}$ and $U_{2_i} = k_{2_i}P$.
  2. Compute $r_i' = r_i \cdot e(U_{2_i}, P)$, if $H_2(m_i, r_i')$ is defined, then abort it. Otherwise sets $H_2(m_i, r_i) = h_i$.
  3. finally, set $V_i = l_iP + k_{2_i}H_3(U_{2_i}, r)Q_A$
  4. Add $(l_i, h_i, k_{2_i}, V_i, U_{2_i}, r_i)$ to VSignature list and return $(V_i, U_{2_i}, r_i)$.
- Otherwise, $\mathcal{F}$ produces the VEsignature by **VE-Sign** algorithm of our proposed scheme. Since he has the private key $as_iP = aH(ID_i)$ of the user with the identity $ID_i$.

**Adjudicator Oracle:** when requesting $(r_i, V_i, U_{2_i}, ID_i, Q_A)$ to adjudicator oracle, if $i \neq X$, $\mathcal{F}$ computes as follows:

1. randomly pick a $k_i \in Z_q$ and compute $r_{1_i} = e(P, k_iP)$.
2. compute $h_i = H_2(m_i, r_{1_i})$ and $U_i = h_iD_X + k_iP$ where $D_X = s_i \cdot aP$

(Note that $\mathcal{F}$ has the private key of the user with the identity $ID_i$) and return $(U_i, r_{1_i}, h_i)$. Otherwise, he aborts it.
**Output:** Finally, $\mathcal{F}$ outputs the original signature $(U^*, r_1^*, h^* = H_2(M, r_1^*))$ under the user with identity $ID_X$ and the adjudicator's public key $Q_A$.

By replays with the same random tape but different choice of oracle $H_2(\cdot)$, as done in the Forking Lemma[10], we can obtain two valid signatures $(m^*, U^*, r_1^*, h^*)$ and $(m^*, U'^*, r_1^*, h'^*)$ which satisfy

$$r^* = e(U^*, P) \cdot e(H_1(ID_X), P_{Pub})^{-h^*} = e(U'^*, P) \cdot e(H_1(ID_X), P_{Pub})^{-h'^*}$$

Since $H_1(ID_X) = aP, P_{Pub} = bP$, we can obtain

$$\begin{aligned} e(U^*, P) \cdot e(H_1(ID_X), P_{Pub})^{-h^*} = e(U'^*, P) \cdot e(H_1(ID_X), P_{Pub})^{-h'^*} &\iff \\ e(U^*, P) \cdot e(aP, bP)^{-h^*} = e(U'^*, P) \cdot e(aP, bP)^{-h'^*} &\iff \\ e(U^* - U'^*, P) = e(aP, bP)^{h^* - h'^*} &\iff \\ abP = \frac{1}{h^* - h'^*}(U^* - U'^*) \end{aligned}$$

Then, given $(aP, bP)$, we can compute $abP$. However, it is well-known that the CDHP is a hard problem. Thus, it means that our proposed scheme is secure against extraction. □

## 7 Conclusion

Verifiably encrypted signatures are the special extension of general signature primitive, and are often used in online contract signing to provide fair exchange. It plays an important role in the e-commerce. In this works, we first show that Gu *et.al* scheme is insecure, then we propose a novel verifiably encrypted signature and prove our proposed scheme to be secure in random oracle model. It is an open problem how to construct a ID-based verifiably encrypted signature without random oracle model.

## References

1. C.X Gu and Y.F Zhu, An ID-based Verifiable Encrypted Signature Scheme Based on Hess's Scheme,CISC 2005, LNCS 3822, pp 42-52, springer-verlag, 2005
2. M.Choudary Gorantla and Ashutosh Saxena, Verifiably Encrypted Signature Without Radom Oracles,ICDCIT 2005, LNCS 3816, pp 357-363, springer-verlag, 2005
3. Asokan,N., Shoup, V., Waidner,M. Optimistic Fair Exchange of Digital Signature (extended abstract), In: Advances in Cryptology-Eurocrypt'98. LNCS 1403, pp 591-606, springer-verlag, 1998
4. Ateniese.G, Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures, In: Proc. of the 6th Conference on CCS.ACM Press pp 138-146 1999
5. Bao. F,Deng, R.H, and Mao.W, Efficient and Practical fair exchange protocols with off-line TTP. In IEEE Symposium on Security and Privacy, Oakland, CA, 1998
6. J.Camenisch and Victor Shoup, Practical Verifiable Encryption and Decryption of Discrete Logarithms, CRYPTO 2003, LNCS 2729, pp 126-144, Springer-verlag, 2003
7. G.Ateniese, Verifiable Encryption of Digital Signature and Applications, ACM Transactions on Information and System Security, Vol7, NO.1, February 2004, pp 1-20
8. Boneh, D., Gentry, C.,Lynn, B., Shacham, H.,Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, In: Advances in Cryptolog-Eurocrypt'03 LNCS 2656, Springer-verlag pp 416-432, 2003
9. Zhang, F., Safavi-Naini, R., Susilo, W., Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings. In: Progess in Cryptology-Indocrypt'03, LNCS 2904, pp 191-204 Springer-verlag, 2003
10. D.Pointcheval and J.Stern,Security Arguments for Digital Signatures and Blind Signatures,Journal of Cryptology, Volume 13 - Number 3,Pages 361-396.
11. Hess,F. Efficient Identity Based Signature Schemes Based on Pairings, In Advances in Cryptology-CRYPTO 2001, LNCS 2139, Springer-verlag, pp 213-229, 2001.