On Security Notions for Verifiable Encrypted Signature

Xu an Wang Xiaoyuan Yang Yiliang Han
Key Laboratory of Information and Network Security
Department of Electronic Technology, Engineering College of Armied Police Force
Xi'an 710086
Wangxahq@yahoo.com.cn

Abstract: First we revisit the security notions for verifiable encrypted signature scheme. We find that the notion of existential unforgeable is not sufficient for fair exchange protocols in most circumstances. And then we reconsider three verifiable encrypted signature schemes in [2][3][6], which we denote as BGLS, MBGLS and GZZ schemes. We find that they are not strong unforgeable. So we propose new improved constructions denoted as NBGLS, MBGLS and NGZZ which are strong unforgeable. Also we reconsider other two verifiable encrypted signature schemes in [4][8], which we denote as ZSS and CA schemes, we find that they cannot resist replacing public key attack. So adjudicator knowing its private key is critical for security in some scenario. At last, We strongly suggest that strong unforgeable for verifiable encrypted signature be a better notion than existential unforgeable and checking adjudicator knowing its private key is a necessary step for secure verifiable encrypted signature scheme.

Keywords: verifiable encrypted signature scheme, security notions

1. Introduction

Fair signature exchange protocol plays an important role in Ecommerce, especially in digital contract signing and e-payment. Generally, there are two main approaches for achieving fair exchange. The first approach is to ensure that the exchange occurs simultaneously, such as having the participants exchange information bit by bit in an interleaving way. The second approach is to ensure the exchange will be completed even though one of the participants refuses to continual. Fair exchange protocols which employ this approach often use a trusted third party (TTP) to store the details of transaction. These details are released if one of the entities refuses to complete the protocol. The use of the online TTP greatly reduces the efficiency of the protocol. Thus optimistic fair exchange protocols based on off-line TTP are more preferable.

In Eurocypt 98 Asokan et al introduced formally a fair exchange protocol relying on a trusted third party (TTP) in an optimistic way [1], but it was not efficient. In Eurocypt 2003, Boneh et al. first proposed a non-interactive verifiable encrypted signature, which is usually used as a building block when constructing optimistic fair exchange, via aggregation of short signatures called BLS scheme [11] based on the bilinear pairing on a gap Diffie-Hellman group (GDH group) [2].Later, Hess presented an attack on [2] and extended its security model and give a new provable secure scheme [3]. In Indocrypt 2003, Zhang et al. presented a new verifiably encrypted signature scheme based on their signature scheme [4].All of the work introduced above are in traditional certificate-based PKI settings, there are also many papers on this topic in the ID-based public key cryptography (ID-PKC). In ICICS 2005, Z. Zhang et al. gave a provably secure optimistic fair exchange protocol based

on SOK-IBS [5]. In CIS05 Gu and Zhu proposed an ID-based verifiably encrypted signature scheme [6] and later they proposed another ID-based verifiably encrypted signature schemes in CISC05 [7]. In ICDCIT05 Choudary and Ashutosh proposed a verifiably encrypted signature scheme provable secure without random oracle [8] .In 2006, J. Zhang and Zou presented a forgery on Gu and Zhu's ID-VESS [7], In addition, they also proposed a verifiably encrypted signature (VES) scheme the size of which is shorter than that of Gu and Zhu [9].

This paper is organized as the following: In section 2, we revisit the security notions for verifiable encrypted signature scheme，especially giving attention on existential unforgeability. We conclude that strong unforgeability is a necessary condition for most applications. Then we cryptanalysis of three verifiable encrypted signature schemes in the strong unforgeability sense [2] [3] [6], give improved VESS and analysis their security. In section 3, we cryptanalysis of other two VESS [4] [8] by replacing public key attack. In section 4, we give our conclusion.

2. Strong Unforgeability VS. Existential Unforgeability for Verifiable Encrypted Signature Scheme

2.1 Security notions for Verifiable Encrypted Signature Scheme

According to [2], a verifiably encrypted signature scheme comprises 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. The algorithms are described below. We refer to the trusted third party as the adjudicator.

Key Generation, **Signing**, **Verification**. As in standard signature schemes.

Adjudicator Key. Generate a public-private key pair (APK, ASK) for the adjudicator.

VESig Creation. Given a secret key SK, a message M, and an adjudicator's public key APK, computes (probabilistically) a verifiably encrypted signature ω on M.

VESig Verification. Given a public key PK, a message M, an adjudicator's public key APK, and a verifiably encrypted signature ω, verify that ω is a valid verifiably encrypted signature on M under key PK.

Adjudication. Given an adjudicator's key pair (APK,ASK), a certified public key PK, and a verifiably encrypted signature ω on some message M, extract and output σ, an ordinary signature on M under PK.

From now on, we denote verifiable encrypted signature scheme as VESS, and we revisit the security notions for VESS.Besides the ordinary notions of signature security in the signature component, they define security properties of VESS: validity, existential unforgeability, and opacity.

Table1 Security notions for VESS in [2]

Validity	VESigVerify(M,VESigCreate(M))=1; Verify(M,Adjudicate(VESigCreate(M))=1
Existential Unforgeability	$\mathrm{Adv}\ \mathrm{VSigF}_F \overset{\mathrm{def}}{=} \Pr\left[\begin{array}{c}\mathrm{VESigVerify}(PK,APK,M,\omega)=\mathrm{valid}\\ (PK\ SK)\overset{R}{\leftarrow}\mathrm{KeyGen}\\ (APK\ ASK)\overset{R}{\leftarrow}\mathrm{AdjKeyGen}\\ (M,\omega)\overset{R}{\leftarrow}F^{S,A}(PK,APK)\end{array}\right]$ Adversary has access to a verifiable encrypted signature creation Oracle S and an adjudication Oracle A along with a hash Oracle, its forgery on M is restricted to not previously being queried to either Oracle.
Opacity	$\mathrm{Adv}\ \mathrm{VSigE}_E \overset{\mathrm{def}}{=} \Pr\left[\begin{array}{c}\mathrm{Verify}(PK,M,\sigma)=\mathrm{valid}\\ (PK\ SK)\overset{R}{\leftarrow}\mathrm{KeyGen}\\ (APK\ ASK)\overset{R}{\leftarrow}\mathrm{AdjKeyGen}\\ (M,\sigma)\overset{R}{\leftarrow}E^{S,A}(PK,APK)\end{array}\right]$ Adversary has access to a verifiable encrypted signature creation Oracle S and an adjudication Oracle A along with a hash Oracle, its forgery on M is restricted to not previously being queried to adjudication Oracle A.

2.2 On Existential Unforgeability

[2] think existential unforgeability is a good security notion for VESS, but we think that's not enough for many applications.

Consider this scenario: in a bank's e-payment system, one user A pays for another user B's good. B requests A transfer 10000 dollars into his count. And then he gives A the good whose value is 10000 dollars. If and only if the forward rounds are completed, the next round begins. A's signature on "Transfer from A's account 10000 dollar to B's account" is a proof for Bank transferring money from A's account to B's account. We use VESS in this scenario. Obviously, Existential Unforgeability is not enough. If one obtains a VESS signature on "Transfer from A's account 10000 dollars to B's account", and he can forge another VESS on the same message, then he can pretend as A! He can get good by transferring A's money to B's account! Such scenarios are very common in applications. So we suggest strong unforgeablity as being a proper security notion for VESS.

2.3 BGLS Scheme and NBGLS Scheme

Now let's revisit the first VESS proposed by Boneh et al based on BLS signature, which we denote as BGLS scheme, and then we give two attacks on this scheme in the strong unforgeable sense.

Table2 BGLS scheme

KeyGen **AdjKeyGen**	The user chooses a random $a \in Z_p$ and compute $v \leftarrow g^a$. The public key is $v \in G$ and the secret key is $a \in Z_p$; The adjudicator chooses a random $b \in Z_p$ and compute $v' \leftarrow g^b$ .The public key is $v' \in G$ and the secret key is $b \in Z_p$.
Sign **Verify**	Given a message $M \in M$ and a secret key a, compute $h \leftarrow H(M)$ and $\sigma \leftarrow h^a$ .The signature is $\sigma \in G$; Given a message $M \in M$, a signature $\sigma \in G$ and a public key $v \in G$, compute $h \leftarrow H(M)$ and output accept if $e(g,\sigma) = e(v,h)$, reject otherwise.
VESigCreate	Input is the message $M \in M$, the user secret key $a \in Z_p$ and adjudicator public key $v \in G$ .Output is the VESS–signature $(\mu,\omega) \in G \times G$ which is computed as follows. Let $h \leftarrow H(M)$ and $\sigma \leftarrow h^a$ Select random $s \in Z_p$ and compute $\mu \leftarrow g^s$ and $\omega \leftarrow \sigma v'^s$ .The VESS-signature is $(\mu,\omega) \in G \times G$.
VESigVerify	Input is the message $M \in M$, $(\mu,\omega) \in G \times G$,the user public key $v \in G$ and the adjudicator public key $v' \in G$ Output is accept if (μ,ω) is a valid VESS-signature on M under v and v' ,that is if $e(g,\omega) = e(v,h) \cdot e(v',u)$ with $h \leftarrow H(M)$ .Otherwise output is reject.
Adjudicate	Input is the message $M \in M$, data $(\mu,\omega) \in G \times G$, the user public key $v \in G$, and the adjudicator public key $v' \in G$ and private key $b \in Z_p$. Output is reject, if VESigVerify rejects M, (μ,ω), v, v', Otherwise output is $\sigma \leftarrow \omega / \mu^b$, which is the ordinary signature on M under v.

Table3 Two attacks on BGLS in strong unforgeability sense

Attack Ⅰ	Attacker gets an ordinary signature $\sigma \in G$, he select random $s \in Z_p$ and compute $\mu \leftarrow g^s$ and $\omega \leftarrow \sigma v'^s$ .The forged VESS-signature is $(\mu,\omega) \in G \times G$ .This means only the signer can transfer an ordinary signature into a VESS-signature.
Attack Ⅱ	Attacker gets valid VESS-signature $(\mu,\omega) \in G \times G$, he select random $r \in Z_p$ and compute $\mu' \leftarrow u g^r$ and $\omega' \leftarrow \omega v'^r$ .The forged VESS-signature is $(\mu',\omega') \in G \times G$. This means we cannot forge a valid VESS-signature from a given valid VESS-signature

The MBGLS scheme proposed in [3] is different from [2] by replacing $h \leftarrow H(M)$ as $h \leftarrow H(M,v)$, so it also suffers from the above two attacks in the strong unforgeable sense.

In order to resist these attacks, we propose a new VESS based on BLS signature. We denote it as NBGLS.

Table4 NBGLS scheme

KeyGen **AdjKeyGen**	The user chooses a random $a \in Z_p$ and compute $v \leftarrow g^a$. The public key is $v \in G$ and the secret key is $a \in Z_p$. The adjudicator chooses a random $b \in Z_p$ and compute $v' \leftarrow g^b$ .The public key is $v' \in G$ and the secret key is $b \in Z_p$ .The adjudicator chooses another $t \in G$, and compute $v'' \leftarrow t^b$. (t, v'') are public parameters.
Sign **Verify**	Same as Table 2 except replacing $H(M)$ by $H(M, v)$.
VESigCreate	Input is the message $M \in M$, the user secret key $a \in Z_p$ and adjudicator public key $v \in G$ and public parameters. Output is the VESS-signature $(\mu, \omega) \in G \times G$ which is computed as follows. Let $h \leftarrow H(M, v)$ and check if $h = v^*$ or $h = t$ .if they do not hold then compute $\sigma \leftarrow h^a$, else return "reject". Select random $s \in Z_p$,and compute $\mu \leftarrow g^s$, $x \leftarrow t^{sa}$ and $\omega \leftarrow \sigma (v')^s (v^*)^{sa}$ .The VESS–signature is (μ, x, ω) .
VESigVerify	Input is the message $M \in M$, $(\mu, \omega) \in G \times G$, the user public key $v \in G$ and the adjudicator public key $v' \in G$ Output is accept if (μ, x, ω) is a valid VESS-signature on M under v and v' ,that is if $e(g, \omega) = e(v, h) \bullet e(v', u) \bullet e(x, v'')$ with $h \leftarrow H(M, v)$ .Otherwise output is reject.
Adjudicate	Input is the message $M \in M$, data (μ, x, ω) , the user public key $v \in G$, and the adjudicator public key $v' \in G$ and private key $b \in Z_p$. Output is reject, if VESigVerify rejects M, (μ, ω) , v , v' , Otherwise output is $\sigma \leftarrow \omega / (\mu x)^b$, which is an ordinary signature on M under v .

First we verify its correctness:

$$\begin{aligned} e(g, \omega) &= e(g, h^a g^{bs} t^{sab}) \\ &= e(g, h^a g^{bs}) e(g, t^{sab}) \\ &= e(g^a, h) e(g^b, g^s) e(g^b, t^{sa}) \\ &= e(v, h) e(v', u) e(v', x) \end{aligned}$$

So VESigVerify (M, VESigCreate (M)) =1, and

$$\omega / (ux)^b = h^a g^{bs} t^{asb} / g^{bs} t^{asb} = h^a$$

It is a valid ordinary signature on M, so Verify (M, Adjudicate (VESigCreate (M)) =1.

Impossible to forge VESS from ordinary signature: Attacker gets σ , his goal is to construct (μ, x, ω) .Obviously he needs to know a , and this is a DLP problem.

Impossible to forge VESS from old VESS signature: Attacker gets $\mu = g^s$, $x = t^{sa}$, $\omega = \sigma (v')^s (v^*)^{sa}$,his goal is to construct $\mu' = g^{s'}$, $x' = t^{s'a}$, $\omega = \sigma (v')^{s'} (v^*)^{sa}$ .Obviously, he needs to know a , and this is also a DLP problem.

NOTE: In **VEsigCreate**, we check if $h = v^*$ or $h = t$, the purpose of this operation is to

resist leaking v'^{a} or t^{a} to adversary.

2.4 GZZ Scheme and NGZZ Schme

We revisit the VESS scheme in [6] which we denote as GZZ scheme and also give two attacks on this scheme in strong unforgeability sense.

Table5 GZZ scheme

KeyGen **AdjKeyGen**	Given $G_1, G_2, q, \hat{e}, P$, return the system parameters $\Omega = (G_1, G_2, q, \hat{e}, P_{pub}, P_a, H_1, H_2)$, the PKG's private key $s \in Z_{q^*}$ and the adjudicator's private key $s_a \in Z_q^*$, where $P_{pub} = sP$, $P_a = s_a P$, $H_1 : \{0,1\}^* \to G_1^*$ and $H_2 : \{0,1\}^* \times G_1 \to Z_q$ are hash functions. Given an identity $ID \in \{0,1\}^*$, compute $D_{ID} = sQ_{ID}$, $Q_{ID} = H_1(ID) \in G_1^*$ .PKG uses this algorithm to extract the user secret key D_{ID} , and gives D_{ID} to the user by a secure channel.
Sign **Verify**	Given a private key D_{ID} and a message m , pick $r \in Z_q^*$ at random, compute, $U = rP$, $h = H_2(m, U)$, $V = rQ_{ID} + hD_{ID}$, and output a signature (U, V) .Given a signature (U, V) of an identity ID for a message m, compute $h = H(m, U)$, accept the signature and return 1 if and only if $\hat{e}(P, V) = \hat{e}(U + hP_{pub}, H_1(ID))$.
VESigCreate	Given a secret key D_{ID} and a message m , choose $r_1, r_2 \in Z_q^*$ at random, compute $U_1 = r_1 P$, $h = H_2(m, U_1)$, $U_2 = r_2 P$, $V = r_1 H_1(ID) + hD_{ID} + r_2 P_a$, and output a verifiably encrypted signature (U_1, U_2, V) .
VESigVerify	Given a verifiably encrypted signature (U_1, U_2, V) of an identity ID for a message m, compute $h = H_2(m, U_1)$, and accept the signature and return 1 if and only if $\hat{e}(P, V) = \hat{e}(U_1 + hP_{Pub}, H_1(ID)) \cdot \hat{e}(U_2, P_a)$.
Adjudicate	Given the adjudicator's secret key s_a and a valid verifiably encrypted signature (U_1, U_2, V) of an identity ID for a message m, compute $V_1 = V - s_a U_2$, and output the original signature (U_1, V_1) .

Table6 Two attacks on GZZ in strong unforgeability sense

Attack Ⅰ	Attacker gets an ordinary signature (U, V) , he selects random $r_2 \in Z_p$ and compute $U_2 = r_2 P$ and $V' = V + r_2 P_a$ .The forged VESS-signature is (U_1, U_2, V') . **This means only the signer can transfer an ordinary signature into a VESS-signature.**
Attack Ⅱ	Given (U_1, U_2, V) and the system parameters $\Omega = (G_1, G_2, q, \hat{e}, P_{pub}, P_a, H_1, H_2)$, we choose random $r_2' \in Z_q$,compute $U_1' = U_1$, $U_2' = r_2' P + U_2$, $V' = V + r_2' P_a$,and output a forged verifiable encrypted signature (U_1', U_2', V') .**This means we cannot forge a valid**

	VESS-signature from a given valid VESS- signature.

We also propose another new VESS based on GZZ scheme, we denote it as NGZZ.

Table7 NGZZ scheme

KeyGen **AdjKeyGen**	Same as Table 5.
Sign **Verify**	Same as Table 5.
VESigCreate	Given a secret key D_{ID} and a message m, choose $r_1, r_2 \in Z_q^*$ at random, compute $U_1 = r_1 P$, $h = H_2(m, U_1)$, $U_2 = r_2 P$, $U_3 = r_1 P_a$, $U_4 = r_1 r_2 P$, $V = r_1 H_1(ID) + hD_{ID} + r_1 r_2 P_a$, and output a verifiably encrypted signature (U_1, U_2, U_3, U_4, V).
VESigVerify	Given a verifiably encrypted signature (U_1, U_2, U_3, U_4, V) of an identity ID for a message m, compute $h = H_2(m, U_1)$, and accept the signature and return 1 if and only if $\hat{e}(P, V) = \hat{e}(U_1 + hP_{Pub}, H_1(ID)) \bullet \hat{e}(U_2, U_3)$.
Adjudicate	Given the adjudicator's secret key s_a and a valid verifiably encrypted signature (U_1, U_2, U_3, U_4, V) of an identity ID for a message m, compute $V_1 = V - s_a U_4$, and output the original signature (U_1, V_1).

First we verify its correctness:

$$\begin{aligned}\hat{e}(P, V') &= \hat{e}(P, r_1 H_1(ID) + hD_{ID} + r_1 r_2 P_a) \\ &= \hat{e}((r_1 + hs)P, H_1(ID)) \bullet \hat{e}(r_2 P, r_1 P_a) \\ &= \hat{e}(U_1 + hP_{Pub}, H_1(ID)) \bullet \hat{e}(U_2, U_3)\end{aligned}$$

So VESigVerify (M, VESigCreate (M)) =1, and

$$\begin{aligned}\hat{e}(P, V_1) &= \hat{e}(P, r_1 H_1(ID) + hD_{ID} + r_1 r_2 P_a - s_a U_4) \\ &= \hat{e}(P, r_1 H_1(ID) + hD_{ID} + r_1 r_2 s_a P - s_a r_1 r_2 P) \\ &= \hat{e}((r_1 + hs)P, H_1(ID)) \\ &= \hat{e}(U_1 + hP_{Pub}, H_1(ID))\end{aligned}$$

It is a valid ordinary signature on M, so Verify (M, Adjudicate (VESigCreate (M)) =1.

Impossible to forge VESS from ordinary signature: Attacker gets $U = rP$, $h = H_2(m, U)$, $V = rQ_{ID} + hD_{ID}$, his goal is to construct $U_1 = r_1 P$, $h = H_2(m, U_1)$, $U_2 = r_2 P$, $U_3 = r_1 P_a$, $U_4 = r_1 r_2 P$, $V = r_1 H_1(ID) + hD_{ID} + r_1 r_2 P_a$, Obviously, he needs to know r_1 ,which is a DLP problem or a CDH problem.

Impossible to forge VESS from old VESS signature: Attacker gets $U_1 = r_1 P$, $h = H_2(m, U_1)$, $U_2 = r_2 P$, $U_3 = r_1 P_a$, $U_4 = r_1 r_2 P$, $V = r_1 H_1(ID) + hD_{ID} + r_1 r_2 P_a$, his goal is to construct $U_1 = r_1 P$, $h = H_2(m, U_1)$, $U_2 = r_2' P$, $U_3 = r_1 P_a$, $U_4 = r_1 r_2' P$, $V = r_1 H_1(ID) + hD_{ID} + r_1 r_2' P_a$, he also needs to know r_1 ,which is a DLP problem or a CDH problem.

3 On adjudicator

In PKC2007, Dodis et al gave a paper on a the security of optimistic fair exchange in the multi-user setting [10], they give examples of secure optimistic fair exchange in the stand-alone setting which are not secure in the multi-user setting. In this section, we further extend their research. We give examples which are secure in the one-adjudicator setting in the multi-adjudicator setting are no longer secure. We will attack two VESS signatures, one is [4] which we denote as ZSS scheme and the other is [8] which we denote as CA scheme.

3.1 ZSS Scheme and Attack on It

Table8 ZSS scheme

KeyGen **AdjKeyGen**	Generate the system parameters: params= $(G_1, G_2, e, q, \lambda, P, H)$. Pick random $x, x_a \in_R z_q^*$,and compute $P_{pub} = xP$, $P_{pubAd} = x_a P$ .The user and the adjudicator's public keys are P_{pub} , P_{pubAd} .The user and the adjudicator's secret key are x and x_a.
Sign **Verify**	Given a secret key x, and a message m, compute $S = (1/(H(m)+x))P$. Given a public key P_{pub}, a message m, and a signature S, verify if $e(H(m)P + P_{pub}, S) = e(P, P)$.
VESigCreate	Given a secret key $x \in Z_q^*$, a message m, and an adjudicator's public key P_{pubAd}, compute $v = (1/(H(m)+x))P_{pubAd}$. The verifiably encrypted signature for m is v.
VESigVerify	Given a public key P_{pub}, a message m ,an adjudicator's public key P_{pubAd}, and a verifiably encrypted signature v, accept v if and only if the following equation holds: $e(H(m)P + P_{pub}, S) = e(P, P_{pubAd})$.
Adjudicate	Given an adjudicator's public key P_{pubAd} and the corresponding private key $x \in Z_q^*$,a certified public key P_{pub} ,and a verifiably encrypted signature v on some message m, ensure that the verifiably encrypted signature is valid, then output $\sigma = x_a^{-1} v$.

Table9 Attack on ZSS in the multi-adjudicator setting

Attack (Replacing Public key Attack)	Suppose real adjudicator's public key is P_{pubAd}, attacker pretends as an adjudicator and publishes his public key as $P_{pubAd}/2$ .Honest user will give his VESS $v = (1/(H(m)+x)) \cdot P_{pubAd}/2$, the attacker now can extract the ordinary signature as following: He just queries $2v$ to the real adjudicator's Adj (.) Oracle and get the ordinary signature.

3.2 CA Scheme and Attack on It

CA scheme is a VESS which is provable secure in stand model. But it also suffers from the replacing public key attack.

Table10 CA scheme

KeyGen **AdjKeyGen**	Pick a generator $P \in G_1$ and $x, y \in Z^*_p$, randomly. Compute $u = xP$, $v = yP \in G_1$ and $z = e(P,P) \in G_2$. The user's private key is (x, y) and public key is (P, u, v, z) .Similarly, the adjudicator's private key is (x_{Ad}, y_{Ad}) and public key is $(P_{Ad}, u_{Ad}, v_{Ad}, z_{Ad})$.
Sign **Verify**	Given a private key $(x, y) \in z^*_p$ and a message $m \in Z^*_p$, pick a random $r \in Z^*_p$ and compute $1/(x+m+yr) P \in G_1$. Here, $1/(x+m+yr)$ is computed modulo p, In the unlikely event that $x + y + mr = 0$, we try again with a different r .The signature is (σ, r). Given a public key (P, u, v, z), a message $m \in Z^*_p$ and a signature (σ, r), accept the signature as valid if this equation holds and reject otherwise: $e(\sigma, u + mP + rv) = z$ .Actually, this is the short signature without random oracle proposed by Boneh et al[12].
VESigCreate	The signer generates a VES on a message $m \in Z^*_p$ using his private key (x, y) and adjudicator's public key $(P_{Ad}, u_{Ad}, v_{Ad}, z_{Ad})$ as follows: 1. Selects a random $r \in Z^*_p$ 2. Computes $\sigma_{VES} = \frac{1}{x+m+yr}(u_{Ad} + rv_{Ad})$ The VES on the message m is (σ_{VES}, r)
VESigVerify	The verifier checks the validity of the VES $\langle \sigma_{VES}, r \rangle$ on a message m using the signer's public key (P, u, v, z), and adjudicator's public key $(P_{Ad}, u_{Ad}, v_{Ad}, z_{Ad})$. He accepts it, if and only if the following equation holds: $e(\sigma_{VES}, u + mP + rv) = e(u_{Ad} + rv_{Ad}, P)$
Adjudicate	When disputes arise between two participating entities, the adjudicator first ensures that the VES $\langle \sigma_{VES}, r \rangle$ on a message m is valid, by executing the VESVerification phase. Then he extracts the original signature using his private key (x_{Ad}, y_{Ad}) as below: $\sigma = \frac{1}{x_{Ad} + ry_{Ad}} \sigma_{VES}$

Table11 Attack on CA in the multi-adjudicator setting

Attack (Replacing Public Key Attack)	Suppose real adjudicator's public key is $(P_{Ad}, u_{Ad}, v_{Ad}, z_{Ad})$, attacker pretends as another adjudicator and publishes his public key as $(P_{Ad}, 2u_{Ad}, 2v_{Ad}, z_{Ad})$ .Honest user will give his VESS-signature σ_{VES} = $\frac{1}{x+m+yr}\ 2(u_{Ad}+rv_{Ad})$,the attacker now can extract the ordinary signature as following: He just queries $\sigma_{VES}/2$ to the real adjudicator's Adj (.) Oracle and get the ordinary signature.

So we must consider replacing public key attack in VESS. How to resist this attack? The adjudicator must prove to the user that he knows the private key corresponding to the public key. They can run the zero-knowledge proof of proof to achieve this goal, and this will make the VESS very complicated. With the help of Trusted PKG, we can reduce the complexity. In this scenario, the adjudicator just has to prove his knowledge to the PKG instead of proving to every user his knowledge of private key.

4 Conclusions

In this paper, we give some considerations on security notions for VESS. We think that existential unforgeability is not a good security notion for VESS, strong unforgeability is more preferable in most applications. So we suggest that strong unforgeability is adopted as right security notion for VESS instead of existential unforgeability.The first three schemes BGLS, MBGLS and ZGG are not secure in strong unforgeable sense. We give attack to the first three schemes and give improved schemes which are strong unforgeable. Actually, we can divide the VESS into two kinds: one kind is just existential unforgeability and the other kind is strong unforgeability. Schemes in [2][3][6] fall in the first kind and Schemes in [4][8] fall in the second kind. But we note that these new schemes are not efficient and signatures are not short, so our further work is finding efficient schemes and short signatures. And we also note security analysis is simple in section 3 and it does not fall in the framework of provable security, so we must improve it which is also our further work.

In section 3 we give another attack-replacing public key attack- to [4] [8], although it's not a very harmful attack, it is dangerous. So we suggest that checking adjudicator knowing its private key is a necessary step for secure verifiable encrypted signature scheme.

Reference

1. N. Asokan, V. Shoup and M. Waidner, "Optimistic fair exchange of digital signatures," Advances in Cryptology - Proceedings of Eurocrypt'98, LNCS 1403, pp.591-606, Springer-Verlag, 1998; IEEE J. on Selected Areas in Communications,18(4):593-610, 2000.

2. D. Boneh, C. Gentry, B. Lynn and H. Shacham, “Aggregate and verifiably encrypted signatures from bilinear maps” Advances in Cryptology - Eurocrypt’03, LNCS 2656, pp.416-432, 2003.
3. F. Hess “On the security of the verifiably-encrypted signature scheme of Boneh, Gentry, Lynn and Shacham” Information Processing letters, 111-114, (89), 2004
4. F. Zhang, R. Safavi-Naini and W. Susilo, “Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings,” INDOCRYPT’03, LNCS2904, pp.191-204, 2003.
5. Zhenfeng Zhang, Dengguo Feng, Jing Xu and Yongbin Zhou, “Efficient ID-Based Optimistic Fair Exchange with Provable Security,” ICICS 2005, LNCS 3783, pp.14-26, 2005.
6. Chunxiang Gu, Yuefei Zhu, Yajuan Zhang. “An Optimistic Fair Signature Exchange Protocol from Pairings,” CIS2005, LANI 3802, PP. 9-16, 2005
7. Chunxiang Gu and Yuefei Zhu, “An ID-Based Verifiable Encrypted Signature Scheme Based on Hess’s Scheme,” CISC’05, LNCS 3822, pp.42-52, 2005.
8. M. Choudary Gorantla and Ashutosh Saxena. “Verifiably Encrypted Signature Scheme Without Random Oracles,” ICDCIT 2005, LNCS 3816, pp. 357–363, 2005.
9. Jianhong Zhang and Wei Zou, “A Robust Verifiably Encrypted Signature Scheme”, EUC Workshops 2006, LNCS 4097, pp.731-740, 2006.
10. Yevgeniy Dodis, PilJoongLee and Dae Hyun Yum,” Optimistic Fair Exchange in a Multi-user Setting” T. Okamoto and X. Wang (Eds.): PKC 2007, LNCS 4450, pp. 118–133, 2007
11. D. Boneh, A. Lynn and H. Shacham, “Short signatures from the Weil pairing,” Advances in Cryptology - Asiacrypt’01, LNCS 2248, pp.514-532, 2001.
12. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: Advances in Cryptology-Eurocrypt’04. Volume 3027 of LNCS., Springer (2004) 56–73