

On Security Notions for Verifiably Encrypted Signature

Xu-an Wang, Xiaoyuan Yang, Yiliang Han

Key Laboratory of Information and Network Security
Engineering College of Chinese Armed Police Force, P.R. China
wangxahq@yahoo.com.cn

Abstract. First we revisit three - BGLS, MBGLS and GZZ verifiably encrypted signature schemes [2, 3, 6]. We find that they are all not strong unforgeable. We remark that the notion of existential unforgeable is not sufficient for fair exchange protocols in most circumstances. So we propose three new - NBGLS, MBGLS and NGZZ verifiably encrypted signature schemes which are strong unforgeable. Also we reconsider other two - ZSS and CA verifiably encrypted signature schemes [4, 8], we find that they both cannot resist replacing public key attack. So we strongly suggest that strong unforgeable for verifiably encrypted signature maybe a better notion than existential unforgeable and checking adjudicator knowing its private key is a necessary step for secure verifiably encrypted signature scheme.

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 continue. 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 Eurocrypt 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 Eurocrypt 2003, Boneh et al. first proposed a non-interactive verifiably 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, 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 verifiably encrypted signature schemes especially giving attention on existential unforgeability. We conclude that strong unforgeability is a necessary condition for most applications. Then we cryptanalysis of three verifiably 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 Verifiably Encrypted Signature Scheme

2.1 Security notions for Verifiably Encrypted Signature Scheme

Definition 1. 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.

- *KeyGen*, *Sign*, and *Verify*: As in standard signature schemes.
- *Adjudicator KeyGen*: 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 w on M .
- *VESig Verification*: Given a public key PK , a message M , an adjudicator's public key APK , and a verifiably encrypted signature w , verify that w 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 w on some message M , extract and output s , an ordinary signature on M under PK .

From now on, we denote verifiably encrypted signature scheme as VESS, and we revisit the security notions for VESS.

Definition 2. Besides the ordinary notions of signature security in the signature component, they define security properties of VESS: validity, existential unforgeability, and opacity.

- **Validity:** $VESigVerify(M, VESigCreate(M)) = 1$; $Verify(M, Adjudicate(VESigCreate(M))) = 1$.
- **Existential Unforgeability:** $(PK, SK) \leftarrow KeyGen, (APK, ASK) \leftarrow AdjKeyGen, (M, w) \leftarrow F^{S,A}(PK, APK), AdvVsigE_F = Pr[VESigVerify(PK, APK, M, w) = valid]$. Adversary has access to a verifiably 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:** $(PK, SK) \leftarrow KeyGen, (APK, ASK) \leftarrow AdjKeyGen, (M, s) \leftarrow E^{S,A}(PK, APK), AdvVsigE_E = Pr[Verify(PK, M, s) = valid]$. Adversary has access to a verifiably 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 price are 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 dollars 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 unforgeability* be 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:

1. **KeyGen, AdjKeyGen :** The user chooses a random $a \in \mathbb{Z}_p$ and compute $v \leftarrow g^a$. The public key is $v \in \mathbb{G}$ and the secret key is $a \in \mathbb{Z}_p$; The adjudicator chooses a random $b \in \mathbb{Z}_p$ and compute $v' = g^b$. The public key is $v' \in \mathbb{G}$ and the secret key is $b \in \mathbb{Z}_p$.

2. **Sign,Verify**: Given a message M and a secret key a , compute $h = H(M)$ and $\sigma = h^a$. The signature is $\sigma \in \mathbb{G}$; Given a message $M \in \mathcal{M}$, a signature $\sigma \in \mathbb{G}$ and a public key $v \in \mathbb{G}$, compute $h = H(M)$ and output accept if $e(g, \sigma) = e(v, h)$, reject otherwise.
3. **VESigCreate**: Input is the message $M \in \mathcal{M}$, the user secret key $a \in \mathbb{Z}_p$ and adjudicator public key $v \in \mathbb{G}$. Output is the VESS signature $(u, w) \in \mathbb{G} \times \mathbb{G}$ which is computed as follows. Let $h = H(M)$ and select random $s \in \mathbb{Z}_p$, compute $u = g^s$ and $w = \sigma v^s$. The VESS signature is $(u, w) \in \mathbb{G} \times \mathbb{G}$.
4. **VESigVerify**: Input is the message $M \in \mathcal{M}$, $(u, w) \in \mathbb{G} \times \mathbb{G}$, the user's public key v and the adjudicator's public key v' . Output is accept if (u, w) is a valid VESS signature on M under v and v' , that is, if $e(g, w) = e(v, h) * e(v', u)$ where $h = H(M)$. Otherwise output is reject.
5. **Adjudicate**: Input is the message $M \in \mathcal{M}$, $(u, w) \in \mathbb{G} \times \mathbb{G}$, the user's public key v , and the adjudicator's public key v' and private key $b \in \mathbb{Z}_p$. If **VESigVerify** rejects M , $(u, w), v, v'$, output is reject, otherwise output is $\sigma = \frac{w}{u^b}$, which is the ordinary signature on M under v .

And then we give two attacks on this scheme in the strong unforgeable sense.

- **Attack ♣**: Attacker gets an ordinary signature $\sigma \in \mathbb{G}$, he selects random $s \in \mathbb{Z}_p$, compute $u = g^s$ and $w = \sigma v^s$. The forged VESS signature is $(u, w) \in \mathbb{G} \times \mathbb{G}$.
- **Attack ♠**: Attacker gets valid VESS signature $(u, w) \in \mathbb{G} \times \mathbb{G}$, he selects random $r \in \mathbb{Z}_p$, computes $u' = ug^r$ and $w' = wv^r$. The forged VESS signature is $(u', w') \in \mathbb{G} \times \mathbb{G}$.

The MBGLS scheme proposed in [3] is different from [2] by replacing $h = H(M)$ as $h = H(M, v)$, 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 signature scheme based on BGLS, We denote it as NBGLS.

1. **KeyGen,AdjKeyGen**: The user chooses a random $a \in \mathbb{Z}_p$ and compute $v \leftarrow g^a$. The public key is $v \in \mathbb{G}$ and the secret key is $a \in \mathbb{Z}_p$; The adjudicator chooses a random $b \in \mathbb{Z}_p$ and compute $v' = g^b$. The public key is $v' \in \mathbb{G}$ and the secret key is $b \in \mathbb{Z}_p$. The adjudicator chooses another generator $t \in \mathbb{G}$, and compute $v'' = t^b$. (t, v'') are public parameters.
2. **Sign,Verify**: Same as Table 2 except replacing $h = H(M)$ by $h = H(M, v)$.
3. **VESigCreate**: Input is the message $M \in \mathcal{M}$, the user secret key $a \in \mathbb{Z}_p$ and adjudicator public key $v \in \mathbb{G}$. Output is the VESS signature $(u, w) \in \mathbb{G} \times \mathbb{G}$ which is computed as follows. Let $h = H(M, v)$ and check if $h = v''$ or $h = t$. if they do not hold then compute $\sigma = h^a$, else return "reject". Select random $s \in \mathbb{Z}_p$, and compute $u = g^s$, $x = t^{sa}$ and $w = \sigma(v')^s(v'')^{sa}$. The VESS signature is (u, x, w) .
4. **VESigVerify**: Input is the message $M \in \mathcal{M}$, $(u, x, w) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}$, the user's public key v and the adjudicator's public key v' . Output is accept if (u, w) is a valid VESS signature on M under v, v', v'' , that is $e(g, w) = e(v, h) * e(v', u) * e(x, v'')$ with $h = H(M, v)$. Otherwise output is reject.

5. **Adjudicate:** Input is the message $M \in \mathcal{M}$, $(u, x, w) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}$, the user's public key v , and the adjudicator's public key v' and private key $b \in \mathbb{Z}_p$. If VESigVerify rejects $(M, (u, x, w), v, v')$, output is reject, otherwise output is $\sigma = \frac{w}{(ux)^b}$, which is the ordinary signature on M under v .

First we verify its correctness:

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

So $\text{VESigVerify}(M, \text{VESigCreate}(M)) = 1$, and

$$\begin{aligned}
\frac{w}{ux^b} &= \frac{h^a g^{bs} t^{asb}}{g^{bs} t^{asb}} \\
&= h^a
\end{aligned}$$

So $\text{Verify}(M, \text{Adjudicate}(\text{VESigCreate}(M))) = 1$.

Security Analysis

- *Impossible to forging VESS from ordinary signature:* Attacker gets σ , his goal is to construct (u, x, w) . Obviously he needs to know a , and this is a DLP problem.
- *Impossible to forging VESS from old VESS signature:* Attacker gets $u = g^s$, $x = t^{sa}$ and $w = \sigma(v')^s (v'')^{sa}$, his goal is to construct $u = g^{s'}$, $x = t^{s'a}$ and $w = \sigma(v')^{s'} (v'')^{sa}$. Obviously, he needs to know a , and this is also a DLP problem.
- *Impossible to leaking v''^a or t^a to adversary:* 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 GZSS scheme and NGZSS schme

We revisit the VESS scheme in [6] which we denote as GZZ scheme:

1. **KeyGen, AdjKeyGen:** Given $\mathbb{G}_1, \mathbb{G}_2, q, e, p$, return the system parameters $\mathbb{G}_1, \mathbb{G}_2, q, e, P_{pub}, P_a, H_1, H_2$, the PKG's private key $s \in \mathbb{Z}_q^*$ and the adjudicator's private key $s_a \in \mathbb{Z}_q^*$, where $P_{pub} = sP, P_a = sP, H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ are hash functions. Given an identity $ID \in \{0, 1\}^*$, computes $D_{ID} = sQ_{ID}, Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$. PKG uses this algorithm to extract the user secret key D_{ID} , and gives D_{ID} to the user by a secure channel.

2. **Sign,Verify**: Given a private key D_{ID} and a message m , pick $r \in \mathbb{Z}_q^*$ at random, compute $U = rP$, $h = H_2(m, U)$, $V = rQ_{ID} + hD_{ID}$, and output a signature (U, V) . Given a signature of an identity ID for a message m , compute $h = H_2(m, U)$, accept the signature and return 1 if and only if $e(P, V) = e(U + hP_{pub}, H_1(ID))$.
3. **VESigCreate**: Given a secret key D_{ID} and a message m , choose $r_1, r_2 \in \mathbb{Z}_q^*$ at random, compute $U_1 = r_1P$, $h = H_2(m, U_1)$, $U_2 = r_2P$, $V = r_1H_1(ID) + hD_{ID} + r_2P_a$, and output a verifiably encrypted signature (U_1, U_2, V) .
4. **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 $e(P, V) = e(U_1 + hP_{pub}, H_1(ID)) * e(U_2, P_a)$.
5. **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 , computes $V_1 = V - s_a U_2$, and outputs the original signature (U_1, V_1) .

And then we give two attacks on this scheme in the strong unforgeable sense.

- **Attack ♣**: Attacker gets an ordinary signature (U, V) , he selects random $r_2 \in \mathbb{Z}_p$ and compute $U_2 = r_2P$ and $V' = V + r_2P_a$. The forged VESS signature is (U_1, U_2, V') .
- **Attack ♠**: Given (U_1, U_2, V) and the system parameters $\mathbb{G}_1, \mathbb{G}_2, q, e, P_{pub}, P_a, H_1, H_2$, we choose random $r'_2 \in \mathbb{Z}_q$, computes $U'_1 = U_1, U'_2 = r'_2P + U_2, V' = V + r'_2P_a$, and output a forged verifiably encrypted signature (U'_1, U'_2, V') .

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

1. **KeyGen, AdjKeyGen** : Same as the original scheme.
2. **Sign,Verify**: Same as the original scheme.
3. **VESigCreate**: Given a secret key D_{ID} and a message m , choose $r_1, r_2 \in \mathbb{Z}_q^*$ at random, compute $U_1 = r_1P, h = H_2(m, U_1), U_2 = r_2P, U_3 = r_1P_a, U_4 = r_1r_2P, V = r_1H_1(ID) + hD_{ID} + r_1r_2P_a$, and output a verifiably encrypted signature (U_1, U_2, U_3, U_4, V) .
4. **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 $e(P, V) = e(U_1 + hP_{pub}, H_1(ID)) * e(U_2, U_3)$.
5. **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 , computes $V_1 = V - s_a U_4$, and output the original signature (U_1, V_1) .

First we verify its correctness:

$$\begin{aligned}
e(P, V') &= e(P, r_1H_1(ID) + hD_{ID} + r_1r_2P_a) \\
&= e((r_1 + hs)P, H_1(ID))e(r_2P, r_1P_a) \\
&= e(U_1 + hP_{pub}, H_1(ID)) * e(U_2, U_3)
\end{aligned}$$

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

$$\begin{aligned}
e(P, V') &= e(P, r_1 H_1(ID) + h D_{ID} + r_1 r_2 P_a - s_a U_4) \\
&= e(P, r_1 H_1(ID) + h D_{ID} + r_1 r_2 P_a - s_a r_1 r_2 P) \\
&= e((r_1 + h s) P, H_1(ID)) \\
&= e(U_1 + h P_{pub}, H_1(ID))
\end{aligned}$$

So $Verify(M, Adjudicate(VESigCreate(M)) = 1$.

Security Analysis

- *Impossible to forging VESS from ordinary signature:* Attacker gets $U = rP$, $h = H_2(m, U)$, $V = rQ_{ID} + hD_{ID}$, his goal is to construct $U_1 = r_1P$, $h = H_2(m, U_1)$, $U_2 = r_2P$, $U_3 = r_1P_a$, $U_4 = r_1r_2P$, $V = r_1H_1(ID) + hD_{ID} + r_1r_2P_a$, Obviously, he needs to know r_1 , which is a DLP problem or a CDH problem.
- *Impossible to forging VESS from old VESS signature:* Attacker gets $U_1 = r_1P$, $h = H_2(m, U_1)$, $U_2 = r_2P$, $U_3 = r_1P_a$, $U_4 = r_1r_2P$, $V = r_1H_1(ID) + hD_{ID} + r_1r_2P_a$, his goal is to construct $U_1 = r_1P$, $h = H_2(m, U_1)$, $U_2 = r'_2P$, $U_3 = r_1P_a$, $U_4 = r_1r'_2P$, $V = r_1H_1(ID) + hD_{ID} + r_1r'_2P_a$, he also needs to know r_1 , which is a DLP problem or a CDH problem.

3 On Adjudicator

In PKC2007, Dodis et al give 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 CT-RSA2008, Huang et al give another paper on the formal model for multi-user setting security [13]. 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 [?] which we denote as CA scheme.

3.1 ZSS Scheme and Attack on It

1. **KeyGen, AdjKeyGen** : Generate the system $params = (\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, H)$. Pick random $x, x_a \in \mathbb{Z}_q^*$, and compute $P_{pub} = xP, P_{pubadv} = x_aP$. The user and the adjudicator's public keys are x, x_a . The user and the adjudicator's secret key are x and x_a .
2. **Sign, Verify**: Given a secret key x , and a message m , compute $S = (\frac{1}{H(m)+x})P$. Given a public key P_{pub} , a message m , and a signature S , verify if $e(H(m)P + P_{pub}, S) = e(P, P)$.
3. **VESigCreate**: Given a secret key $x \in \mathbb{Z}_q^*$, a message m , and an adjudicator's public key P_{pubadv} , compute $v = (\frac{1}{H(m)+x})P_{pubadv}$. The verifiably encrypted signature for m is v .

4. **VESigVerify**: Given a public key P_{pub} , a message m , an adjudicator's public key P_{pubadv} , and a verifiably encrypted signature ν , accept ν if and only if the following equation holds: $e(H(m)P + P_{pub}, S) = e(P, P_{pubadv})$.
5. **Adjudicate**: Given an adjudicator's public key P_{pubadv} and the corresponding private key $x \in \mathbb{Z}_q^*$, 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 $= x_a^{-1}\nu$.

And then we give replacing public key attack on this scheme .

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

3.2 CA Scheme and Attack on It

1. **KeyGen, AdjKeyGen** : Pick a generator $P \in \mathbb{G}_1$ and $x, y \in \mathbb{Z}_p^*$, randomly. Compute $u = xP$, $v = yP \in \mathbb{G}_1$ and $z = e(P, P) \in \mathbb{G}_2$. 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 $(P_{Ad}, u_{Ad}, v_{Ad}, Z_{Ad})$.
2. **Sign, Verify**: Given a private key $(x, y) \in \mathbb{Z}_p^*$ and a message $m \in \mathbb{Z}_p^*$, pick a random $r \in \mathbb{Z}_p^*$ and compute $(\frac{1}{x+y+mr})P \in \mathbb{G}_1$. Here, $\frac{1}{x+y+mr}$ 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 $m \in \mathbb{Z}_p^*$ and a signature (σ, r) , accept the signature as valid if the equation $e(\sigma, u + mP + rv) = z$ holds and rejects otherwise. Actually, this is the short signature without random oracle proposed by Boneh et al [12].
3. **VESigCreate**: The signer generates a VES on a message $m \in \mathbb{Z}_p^*$ using his private key (x, y) and adjudicator's public key $(P_{Ad}, u_{Ad}, v_{Ad}, Z_{Ad})$ as follows:
 - Selects a random $r \in \mathbb{Z}_p^*$.
 - Computes $\sigma_{VES} = (\frac{1}{x+y+mr})(u_{Ad} + rv_{Ad})$.

The VES on the message m is (σ_{VES}, r) .

4. **VESigVerify**: The verifier checks the validity of the VES (σ_{VES}, r) 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: $e(\sigma_{VES}, u + mP + rv) = e(u_{Ad} + rv_{Ad}, P)$.
5. **Adjudicate**: When disputes arise between two participating entities, the adjudicator first ensures that the VES (σ_{VES}, r) 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 $\sigma = (\frac{1}{x_{Ad} + ry_{Ad}})\sigma_{VES}$:

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

- **Attack ♠:** Suppose real adjudicator's public key is $(P_{Ad}, u_{Ad}, v_{Ad}, Z_{Ad})$, the 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 $\sigma_{VES} = (\frac{1}{x+y+mr})(2(u_{Ad} + rv_{Ad}))$, the attacker now can extract the ordinary signature as following: He just queries $\frac{\sigma_{VES}}{2}$ to the real adjudicator's $Adj(.)Oracle$ and get the ordinary signature.

3.3 Some Remarks

So we must consider replacing public key attack in VESS. How to resist this attack? The adjudicator must prove to the user knowledge of the private key corresponding to his public key. They can run the zero-knowledge proofs of knowledge 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 Conclusion

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 verifiably encrypted signature scheme.

Acknowledgement.

The authors would like to express their gratitude thanks to Dr Xinyi Huang for his suggestions to improve this paper.

References

1. N. Asokan, V. Shoup and M. Waidner. Optimistic fair exchange of digital signatures. In *Eurocrypt 1998*, LNCS 1403, pages 591–606. Springer–Verlag, 1998.
2. D. Boneh, C. Gentry, B. Lynn and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Eurocrypt 2003*, LNCS 2656, pages 416–432. Springer–Verlag, 2003.
3. F. Hess. On the security of the verifiably encrypted signature scheme of Boneh, Gentry, Lynn and Shacham” *Information Processing letters, Vol. 89.*, pages. 111–114, 2004.
4. F. Zhang, R. Safavi-Naini and W. Susilo. Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings. In *Indocrypt 2003*, LNCS 2904, pages 191–204. Springer–Verlag, 2003.
5. Zhenfeng Zhang, Dengguo Feng, Jing Xu and Yongbin Zhou. Efficient ID-Based Optimistic Fair Exchange with Provable Security. In *ICICS 2005*, LNCS 3783, pages 14–26. Springer–Verlag, 2005.
6. Chunxiang Gu, Yuefei Zhu, Yajuan Zhang. An Optimistic Fair Signature Exchange Protocol from Pairings. In *CIS 2005*, LNAI 3802, pages. 9–16. Springer–Verlag, 2005.
7. Chunxiang Gu and Yuefei Zhu. An ID-Based Verifiably Encrypted Signature Scheme Based on Hess’s Scheme. In *CISC 2005*, LNCS 3822, pages 42–52. Springer–Verlag, 2005.
8. M. Choudary Gorantla and Ashutosh Saxena. Verifiably Encrypted Signature Scheme Without Random Oracles. In *ICDCIT 2005*, LNCS 3816, pages. 357–363. Springer–Verlag, 2005.
9. Jianhong Zhang and Wei Zou. A Robust Verifiably Encrypted Signature Scheme. In *EUC 2006*, LNCS 4097, pages. 731–740. Springer–Verlag, 2006.
10. Yevgeniy Dodis, PilJoongLee and Dae Hyun Yum. Optimistic Fair Exchange in a Multi-user Setting. In *PKC 2007*, LNCS 4450, pages. 118–133. Springer–Verlag, 2007.
11. D. Boneh, A. Lynn and H. Shacham. Short signatures from the Weil pairing. In *Asiacrypt 2001*, LNCS 2248, pages. 514–532. Springer–Verlag, 2001.
12. D. Boneh, X. Boyen. Short Signatures Without Random Oracles. In *Eurocrypt 2004*, LNCS 3207, pages. 56–73, Springer–Verlag, 2001.
13. Q. Huang, G. Yang, S. Wong and W. Susilo. Efficient Optimistic Fair Exchange Secure in the Multi-user Setting and Chosen-key Model without Random Oracles. In *CT-RSA 2008*, LNCS 4964, pages. 106–120, Springer–Verlag, 2008.