A provably secure really source hiding designated verifier signature scheme based on random oracle model

Huang-Ta Huang[1], *Jue-Sam Chou[2]
[1]Department of Information Management, Nanhua University Chiayi 622 Taiwan, R.O.C
g6451502@mail1.nhu.edu.tw
[2]Department of Information Management, Nanhua University Chiayi 622 Taiwan, R.O.C
*: corresponding author: jschou@mail.nhu.edu.tw
Tel: 886+ (0)5+272-1001 ext.56536

Abstract.

A lot of designated verifier signature (DVS) schemes have been proposed. However, all of them only provide the basic security requirement that only the designated verifier can check the validity of the signature. They are either not secure enough or lacking source hiding. Hence, in this article, we design a provably secure DVS scheme. It not only can attain the basic security requirement but also hide the original signer's identity which makes our scheme more suitable for the applications in an electronic voting system.

Keyword: *DVS, secure hash functions, random oracle, bilinear pairings, Diffie-Hellman Problem*

1. Introduction.

There are many research works on DVS scheme. In 1996, Jakobsson *et al.* [1] proposed a method of designated verifier signature scheme. In it, the designated verifier could prove the exactness of the signature received from the signer. Then, the designated verifier can imitate the signer to sign the message. He can make the same signature as the signer does so that anyone can't distinguish who was the original signer. Subsequently, many related articles about DVS have been proposed.

In 2003 [2], G. Wang pointed out that Jakobsson *et al.*'s scheme is insecure by illustrating a simple attack that an adversary can convince the designated verifier to receive an invalid signature. In 2004, Laguillaumie *et al.* [3, 4] proposed two schemes: (1) a multi-designated verifier signature [3], and (2) designated verifier signatures: anonymity and efficient construction from any Bilinear Map [4]. However, both of their schemes don't have source hiding property. Since that signer's identity is used by the verifier in the verification phase. In 2006 [5], Lal *et al.* proposes four ID based strong designated verifier proxy signature schemes; however, each doesn't possess the source hiding, neither. In 2007 [6], Laguillaumie *et al.* proposed a multi-designated verifier signature which protects the anonymity of signers without encryption. However, Shim [11] shows that Laguillaumie *et al.*'s scheme [6] is insecure against rogue-key attack. Moreover, we found their scheme doesn't possess source hiding as well since the verifier uses the public key of the signer to verify weather $e(M,P) = e(Q_A, P_A)e(Q_B, P_B)$ holds, where P_A is the signer's public key. In 2008, Kang *et al.* [15] proposed a novel identity-based strong designated verifier signature scheme with two claimed advantages, low communication and computational cost. However, later Du *et al.* [8], in 2008, found an impersonation attack on [15]. Hence, they provided a modification on [15]. They claimed that their scheme achieves all security requirements of strong DVS inducing source hiding. Also in 2008, Zhang *et al.* [9] proposed a novel ID-based DVS. They claimed that their scheme satisfies the property of source hiding. However on the contrary, we found [8, 9] both lack the source hiding property since the verifier in each of them uses of the signer's public key for doing the verification. For example, the verification equation in [8] is $\sigma = e(t + h\underline{Q_A}, d_B)$ and [9] is $e(U_1, V) = e(S_{ID_B}, \underline{Q_{ID_A}})$ (Here and thereafter, we use an underline to indicate the problem part in the verification equation.) Also, in 2008, Lal *et al.* proposed an identity based strong bi-designated verifier proxy signature scheme [7]. In their scheme, only the two designated verifiers can verify whether the proxy signature is signed by the original signer without both being able to transfer this signature to others. That is, both cannot convince the other party that who was the original signer of a given signature. Moreover, they claimed that their scheme is unforgeable. However, we will demonstrate a forgery attack on their protocol in this paper. In 2009, Kang *et al.* proposed two designated verifier signature schemes [14]. They claimed that both of their schemes are strong and unforgeable. Nevertheless, we found that both of their schemes lacks the source hiding since in the first protocol, it uses $U' = \mathrm{r}' Q_{ID_A}$ and $\sigma' = H_2\left(M, e\left(U', S_{ID_C}\right)\right)$ in the signature simulation and the warrant W in the second records the identities of the original signer and proxy signer. Moreover, the second protocol suffers insider forgery attack. We will demonstrate the forgery attack in the second protocol in this paper. Also, in 2009 [17], Cao *et al.* proposed a secure identity based universal DVS scheme in the standard model based on bilinear pairings. However, the way of the bilinear mapping they use is different from the common rule that G_1 is an additive group and G_2 is a multiplicative group. (*e.g.* Common rule: $e(x+y+z, g) = e(x,g)e(y,g)e(z,g)$; Cao *et al.*s': $e(xyz, g) = e(x,g)e(y,g)e(z,g)$). Moreover, it lacks the source hiding as well because of the verification equation $e(A,g) = e(g_2, \underline{g_1})\ e(u' \prod i \in \mathcal{U}, B)\ \ e(m' \prod j \in m, C)$, where g_1 is the signer's public key. Thus, in this article, we will propose a novel DVS that is more secure and really has the anonymity property of signer's identity.

In a DVS scheme, the original signer sends a signature on a message to the designated verifier for the verifier to check the validity of the signature by using his secret key. For the literature we received, we can see that there has existed two cases in the verification and verification phase in the literature: (a) the verifier uses of the signer's public key in both of the verification and simulation phases, he can identify the source of a given message but unable to prove to a third party about the source identity, the related schemes are [1, 8, 9, 10, 12, 13, 15, 16], and (b) the verifier uses signer's public key only in the simulation phase, he can identify the source of a given message without the capability of proving the source identifier to a third party, the related schemes are [7,14]. In this article, we proposed the third case: (c) the verifier needs not use signer's public key in both phases of verification and simulation. This is the reason why our scheme really source hiding property. We will prove its security. We argue that our scheme can resist the conditional KCI attack which we define as follows: Even if the verifier's private key has been compromised by adversary *E*, due to the identity of the original signer cannot be revealed, *E* cannot masquerade as the signer to communicate with the verifier. We will explain why our scheme can resist such a conditional KCI attack in this article.

The remainder of this paper is organized as follows: In Section 2, we introduce some preliminaries. In Section 3, we review and attack on the second protocol proposed by Kang *et al.* [14]. Then, we present a novel scheme in Section 4 and analyze its security in Section 5. The discussions and comparisons are made in Section 6. Finally, a conclusion is given in Section 7.

2. Preliminaries

In this section, we will briefly describe the basic concepts and properties of bilinear pairing and some related problems.

2.1 Bilinear pairings

Let G_1 be a cyclic group generated by P, whose order is a prime q and G_2 be a cyclic multiplicative group of the same order. It is assumed that the discrete logarithm problems (DLP) in both G_1 and G_2 are hard. Let e: $G_1 \times G_1 \rightarrow G_2$ be a pairing which satisfies the following conditions：

Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$,where $a, b \in_R Z_q^*$, $P, Q \in_R G_1$

Non-degenerate: There exists P and $Q \in_R G_1$; $e\,(P, Q) \neq 1$

Computability: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

2.2 Some related problems:

Let G be a cyclic multiplicative group generated by g with prime order q. The definitions of the problems are described as follows.

(1)*Discrete Logarithm Problem (DLP):* Given a couple of elements y and g, find an integer $a \in z_q^*$, such that $y=g^a$.

(2)*Computation Diffie-Hellman Problem (CDHP):* Given (g, g^a, g^b) for a, $b \in z_q^*$, compute g^{ab}.

(3)*Decision Diffie-Hellman Problem (DDHP):* Given (g^x, g^y, g^z) for x, y, $z \in z_q^*$, decide whether $z \equiv xy \ (mod)\ q$.

Thus, if we have an algorithm that can solve *DDHP*, then it can be used to solve *CDHP* and *DLP*. But indeed no such algorithm exists nowadays.

(4)*Elliptic Curve Discrete Logarithm Problem (ECDLP):* Given $P \in G_1$, and xP, where $x \in Z_q^*$. The *ECDL problem* is to find x.

(5)*Bilinear Diffie-Hellman Problem (BDHP):* Given a randomly chosen generator $P \in G_1$, as well as $aP, bP,$ and cP (for unknown random values $a, b, c \in Z_q$), the BDH problem is to compute $e(P,P)^{abc}$ in G_2.

3. Review and attack on Kang et al.'s protocol

Kang *et al.* proposed two protocols [14] for preventing key exposure. However, after analysis, we found that their second protocol still lack source hiding. Moreover, the second protocol suffers from the insider forgery attack. In the following, we first review then show the attack on Kang *et al.'s* second protocol in Section 3.1 and Section 3.2 respectively.

3.1 Review of Kang et al.'s second protocol *(as shown in Figure 1.)*

In their scheme, there exist three people. They are the original signer Alice, proxy signer Bob, and designated verifier Cindy, respectively. In the following, we roughly describe their scheme.

First, Alice picks a random value $r \in Z_q^*$, and then calculates $U = r\ Q_{ID_A}$ and $\sigma = H_2(W, e(rQ_{ID_B}, S_{ID_A}))$, where W is the warrant which records the identities of the original signer and the proxy signer. Alice sends (σ, U, W) to Bob. Bob checks if $\sigma = H_2(W, e(U, S_{ID_B}))$ holds. If the equation holds, Bob produces a proxy signature by selecting a random value $t \in Z_q^*$, and computing $X = tQ_{ID_B}$, $S_{ID_P} = t^{-1}\sigma + S_{ID_B}$, and $V = H_2(M, W, e(tQ_{ID_C}, S_{ID_P}))$, Then, Bob transfers (M, W, σ, X, V) to Cindy.

After receiving the information from Bob, Cindy checks to see whether message *M* confirms to the warrant *W*. If so, Cindy confirms that both Alice and Bob are on the warrant. If the confirmation succeeds, Cindy accepts the signature, if and only if $V = H_2(M, W, e(Q_{ID_C}, \sigma)e(S_{ID_C}, X)$

3.2 Attack on Kang et al.'s second protocol

We found that their scheme suffers the masquerading attack. Since an attacker *E* may camouflage Alice to sends out a signature to Bob by first picking a random value $r \in Z_q^*$ and then computing $U^* = rQ_{ID_A}$, $\sigma^* = H_2(W, e(rQ_{ID_B}, S_{ID_E}))$. He then sends (σ^*, W, U^*) to Bob, where the warrant *W* records both the signer and verifier as ID_A and ID_B rather than ID_E and ID_B. It is obvious that it will pass Bob's verification as shown in figure 2.

4. The proposed scheme

In this section, we present a novel method to get rid of all possible attacks. Our scheme adopts the concept of ID-based cryptography. In the following, we will describe our ID-based designated verifier signature scheme (ID-DVS) and also show it in Figure 3.

Our scheme includes five phases: (1) *Setup*, (2) *Extract*, (3) *SigGen*, (4) *SigVer*, and *(5) SigSim.*

(1) *Setup:*

Let G_1 be an additive cyclic group with a prime order *q*, G_2 be a multiplicative cyclic group of the same order and *P* be a generator of G_1, $H_i(*), i \in \{1, 2\}$, be two cryptographic hash functions with $H_1: \{0,1\}^* \rightarrow G_1$, $H_2: \{0,1\}^* \times G_2 \rightarrow Z_q^*$, *e* be a bilinear map with $e: G_1 \times G_1 \rightarrow G_2$. Then, KGC picks a random value $s \in Z_q^*$ as the system master secret key and calculates the corresponding public key as $P_{pub} = sP$. The system parameter set is $\{G_1, G_2, P, P_{pub}, H, e, q\}$.

(2) *Extract:*

Given a user's identity ID, KGC computes $Q_{ID} = H_1(ID)$, $S_{ID} = sQ_{ID}$ and returns (S_{ID}, Q_{ID}) to the user ID as his private key and public key.

(3) *SigGen:*

①The signer Alice selects a random value $\alpha \in Z_q^*$.

②Computes δ, ε and ξ as follows:

$\delta = \alpha Q_A$

$\varepsilon = e(P_{pub}, Q_B)$
$\xi = H_2(m, \varepsilon)S_A$
③Sends signature $\sigma = e(\xi + S_A, Q_B)^{\alpha}$ and δ to verifier Bob.

(4) ***SigVer:***

After receiving (δ, σ), Bob verifies the validity of the signature by checking whether or not $\sigma = e(\delta, S_B)^{H_2(m,\varepsilon)+1}$ holds. If it doesn't hold, he rejects.

(5) ***SigSim:***

At this stage, Bob can simulate correct signature transcript for message *m* to be verified successfully as follows:

①Bob picks a random value $\beta \in Z_q^*$.
②Bob computes $\tilde{\delta}$ and $\tilde{\sigma}$ as follows.
$\tilde{\delta} = \beta\delta$
$\tilde{\sigma} = e(\tilde{\delta}, S_B)^{H_2(m_i,\varepsilon)+1}$
③The simulated signature is of *m* is $(\tilde{\delta}, \tilde{\sigma})$.

5. Security analysis

In this section, we analyze the security of our scheme. In Settion 5.1, we show that our scheme is correct. In Section 5.2, we assume that an adversary $\mathcal{F}$ can succeed in disguising as either Alice or Bob to sign on his random chosen message m_i; however, we will show that this assumption contradicts to the problem of BDH. In addition, in Section 5.3, we will demonstrate that our scheme possesses the anonymous property for the sender. We show that our scheme has the ability of non-interactive in Section 5.4, possesses the deniable property in Section 5.5, and can be applied to an electronic voting system for its avoidance of conditional KCI attack in Section 5.6. We will give a definition for conditional KCI attack there.

5.1. Correctness

In our scheme, as long as a signature (δ, σ) on message *m* is formed according to our specification, it can be proved correctly by designated verifier Bob using the following equation:

$$\sigma = e(\xi + S_A, Q_B)^{\alpha} = e(H_2(m,\varepsilon)Q_A + Q_A, S_B)^{\alpha} = e(\alpha(Q_A + Q_A), S_B)^{H_2(m,\varepsilon)}$$
$$= e(\alpha Q_A + \alpha Q_A, S_B)^{H_2(m,\varepsilon)} = \sigma = e(\delta, S_B)^{H_2(m,\varepsilon)+1}$$

5.2. Anti-forgeability

Theorem. Suppose that there is an adversary $\mathcal{F}$ who can pretend to be ID_i or ID_j (with each unequal to $\mathrm{ID_A}$ and $\mathrm{ID_B}$) to forge the signature of ID_i and ID_j on message *m* (which can be verified successfully using $\mathrm{ID_A}$ and $\mathrm{ID_B}$), then there must exist an algorithm $\mathcal{B}$ which can solve BDH problem with non-negligible probability.

Proof: If $\mathcal{F}$ exists, then we can construct an algorithm $\mathcal{B}$ to solve bilinear Diffie-Hellman problem after intracting with $\mathcal{F}$ as follows:

Given a BDH instance $(\mathrm{aP}, \mathrm{bP}, \mathrm{cP})$ for randomly chosen $\mathrm{a}, \mathrm{b}, \mathrm{c} \in \mathrm{Z}_\mathrm{q}^*$ with $\mathrm{Q_A} = \mathrm{H_1}(\mathrm{ID_A}) = \mathrm{aP}$, and $\mathrm{Q_B} = \mathrm{H_1}(\mathrm{ID_B}) = \mathrm{bP}$ being the signer's and the designated verifier's public keys respectively and cP being the system public key, $\mathcal{B}$'s goal is to compute $\mathrm{e}(\mathrm{P}, \mathrm{P})^{\mathrm{abc}}$ using the following steps. We also summaries the relative inputs and outputs of algorithm $\mathcal{F}$ and $\mathcal{B}$ in figure 4 and figure 5, respectively.

Step1. $\mathcal{B}$ sets $\mathrm{Q_A} = \mathrm{aP}, \mathrm{Q_B} = \mathrm{bP}$, and $\mathrm{P_{pub}} = \mathrm{cP}$ as the system public key, where *c* is system master secret key, then sends the parameter set $\{\mathrm{G_1}, \mathrm{G_2}, \mathrm{P_{pub}}, \mathrm{H_1}, \mathrm{H_2}\}$ to $\mathcal{F}$, where $\mathrm{H_1}$ and $\mathrm{H_2}$ are two random ora -cles and controlled by $\mathcal{B}$.

Step2. *Key Extract Query:*

$\mathcal{F}$ queries to $\mathrm{H_1}$ with ID_i. $\mathrm{H_1}$ outputs $\mathrm{Q}_i = \mathrm{aP}$ if $\mathrm{ID}_i = \mathrm{ID_A}$, bP if $\mathrm{ID}_i = \mathrm{ID_B}$, $\mathrm{r}_i\mathrm{P}$ otherwis, $\mathrm{r}_i \in \mathrm{Z}_\mathrm{q}^*$ (shown as follows).

$$\mathrm{Q}_i = \mathrm{H_1}(\mathrm{ID}_i) = \begin{cases} \mathrm{aP}, \text{if } \mathrm{ID}_i = \mathrm{ID_A} \\ \mathrm{bP}, \text{if } \mathrm{ID}_i = \mathrm{ID_B} \\ \mathrm{r}_i\mathrm{P}, \text{otherwise, where } \mathrm{r}_i \in \mathrm{Z}_\mathrm{q}^* \text{ chosen by } \mathcal{B} \end{cases}$$

$\boldsymbol{H_1 - query}$ (*public key*): As $\mathcal{F}$ wants to query on ID_i (which is not equal to $\mathrm{ID_A}$ or $\mathrm{ID_B}$), $\mathcal{B}$ looks for $(\mathrm{ID}_i, \mathrm{Q}_i)$ in $\mathrm{H}_1^{\mathrm{list}}$.

1) If $\mathrm{ID}_i \neq \mathrm{ID_A}$ and $\mathrm{ID_B}$, then $\mathcal{B}$ returns $\mathcal{S}_i = \mathrm{r}_i\mathrm{cP}$ as the private key corresponding to $\mathrm{H_1}(\mathrm{ID}_i)$ for ID_i, where cP is $\mathrm{P_{pub}}$, and inserts $(\mathrm{ID}_i, \mathrm{Q}_i, \mathcal{S}_i)$ to $\mathrm{H}_1^{\mathrm{list}}$.

2) Otherwise, $\mathcal{B}$ responses with failure, which means ID_i is equal to $\mathrm{ID_A}$ or $\mathrm{ID_B}$.

Note that the purpose of $\mathcal{F}$ is not to obtain the private key $\mathcal{S}_\mathcal{A} = \mathrm{acP}$ of $\mathrm{ID_A}$ or $\mathcal{S}_\mathcal{B} = \mathrm{bcP}$ of $\mathrm{ID_B}$, it is to set the private key of

[Alice]

random value $r \in Z_q^*$

$U = r\,Q_{ID_A}$

$\sigma = H_2(W, e(rQ_{ID_B}, S_{ID_A}))$

(σ, U, W) → [Bob]

Checks if $\sigma = H_2(W, e(U, S_{ID_B}))$

random value $t \in Z_q^*$

$X = tQ_{ID_B}$

$S_{ID_P} = t^{-1}\sigma + S_{ID_B}$

$V = H_2(M, W, e(tQ_{ID_C}, S_{ID_P}))$

[Cindy]

← (M, W, σ, X, V)

Checks message *M* to warrant *W*

Checks whether Alice and Bob

Accepts if $V = H_2(M, W, e(Q_{ID_C}, \sigma)e(S_{ID_C}, X))$

Figure 1 Kang *et al.*'s second protocol

Adversary[E]

random value $r \in Z_q^*$

computes $U^* = r\,Q_{ID_E}$

$\sigma^* = H_2(W, e(rQ_{ID_B}, S_{ID_E}))$

proxy signer[Bob]

(σ^*, W, U^*) → checks if

$\sigma^* = H_2(W, e(U^*, S_{ID_B}))$

generates

$X = tQ_{ID_B}$

$S_{ID_P} = t^{-1}\sigma^* + S_{ID_B}$

$V = H_2(M, W, e(tQ_{ID_C}, S_{ID_P}))$

designated verifier ← (M, W, σ^*, X, V)

[Cindy]

Checks message *M* to warrant *W*

Checks whether Alice and Bob

Accepts if

$V = H_2(M, W, e(Q_{ID_C}, \sigma^*)e(S_{ID_C}, X))$

Figure 2 Forgery attack on Kang *et al.*'s second protocol

[Alice] [Bob]

s is master secret key

$P_{pub} = sP$

$m \in \{0,1\}^*$, $H_2: \{0,1\}^* \times G_2 \to Z_q^*$

picks a random value $\alpha \in Z_q^*$

$\delta = \alpha Q_A$

$\varepsilon = e(P_{pub}, Q_B)$

$\xi = H_2(m,\varepsilon)S_A$

$\sigma = e(\xi + S_A, Q_B)^{\alpha}$

$\xrightarrow{(\delta, \sigma), m}$

Checks if

$\sigma = e(\delta, S_B)^{H_2(m,\varepsilon)+1}$

then simulates

picks a random value

$\beta \in Z_q^*$

$\tilde{\delta} = \beta\delta$

$\tilde{\sigma} = e(\tilde{\delta}, S_B)^{H_2(m,\varepsilon)+1}$

Figure 3 Our proposed scheme

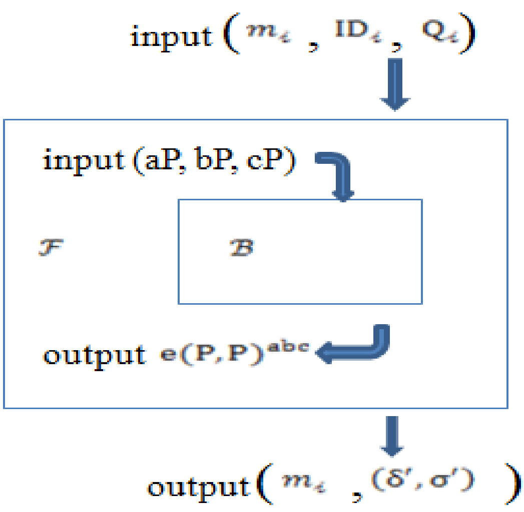

Figure 4 The inputs and outputs in algorithm $\mathcal{F}$ and $\mathcal{B}$.

ID_i or ID_j as $r_i cP$ with $i, j \neq \{A, B\}$, where $r_i \in Z_q^*$ is his random chosen number.

$\boldsymbol{H_2 - query}$: As $\mathcal{F}$ wants to query H_2-Oracle with (m_i, ε_i), $\mathcal{B}$ checks the H_2-list. If $H_2(m_i, \varepsilon_i)$ already exists in the list, he aborts. Else, $\mathcal{B}$ randomly chooses $P_i \in Z_q^*$ and adds the tuple $(m_i, \varepsilon_i, P_i(= H_2(m_i, \varepsilon_i)))$ to the list H_2^{list}.

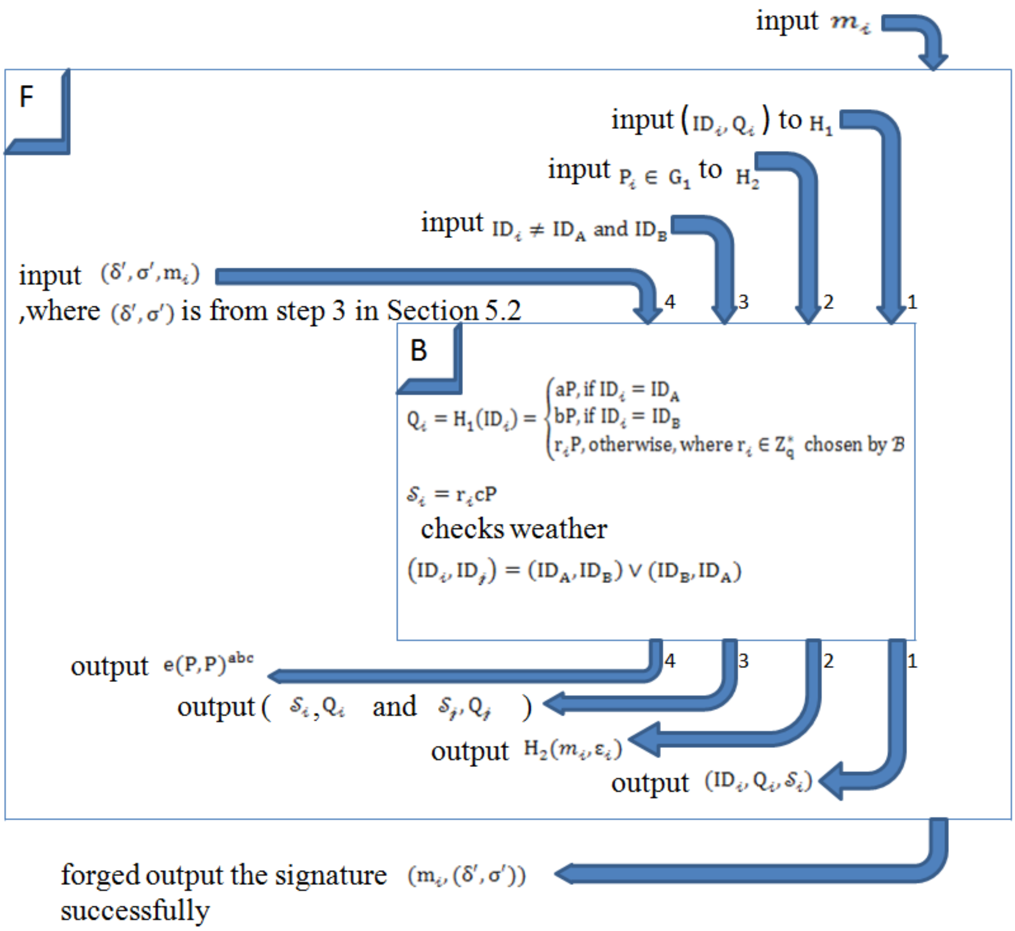

Figure 5 Overall structure of algorithm $\mathcal{B}$ of $\mathcal{F}$

Step3. Signing Query: When adversary $\mathcal{F}$ queries the signature on message m_i (That is, $\mathcal{F}$ pretends to be the signer ID_i for signing m_i.), and sends the signer/designated verifier's identity ID_i/ID_j to $\mathcal{B}$, $\mathcal{B}$ runs as below:

1) Siging: If $ID_i \neq ID_A$ and ID_B, $\mathcal{B}$ will response the private key $\mathcal{S}_i = r_i cP$ of ID_i to $\mathcal{F}$. $\mathcal{F}$ picks a random value $\alpha' \in Z_q^*$ and calculates the parameters by following equations.

$$\delta' = \alpha' Q_i$$
$$\varepsilon' = e(P_{pub}, Q_j)$$
$$\xi' = H_2(m_i, \varepsilon') S_i$$

$$\sigma^{'} = e(\xi^{'} + \mathcal{S}_{i}, Q_{j})^{\alpha^{'}}$$

2) Simulation: If $ID_{j} \neq ID_A$ and ID_B, $\mathcal{B}$ will response the private key $\mathcal{S}_{j} = r_{j}cP$ of ID_{j} to $\mathcal{F}$. Then, $\mathcal{F}$ selects a random value $\alpha^{'} \in Z_q^*$ and calculates the following:

$$\delta^{'} = \alpha^{'} Q_{j}$$
$$\varepsilon^{'} = e(P_{pub}, Q_{j})$$
$$\xi^{'} = H_2(m_{i}, \varepsilon^{'}) S_{j}$$
$$\sigma^{'} = e(\delta^{'}, S_{j})^{H_2(m_{i}, \varepsilon^{'})+1}$$

3) Otherwise, $\mathcal{B}$ aborts and stops this signature forgery.

Finally, $\mathcal{F}$ returns $(\delta^{'}, \sigma^{'})$ as the forgery signature as if it were signed by ID_{i} or ID_{j} on message m_{i}.

Step4. Verifying query: Given the signature $(\delta^{'}, \sigma^{'})$, $\mathcal{F}$ pretends to be ID_{j} the designated verifier for verifying its validity. He calls algorithm $\mathcal{B}$ to check whether $(ID_{i}, ID_{j}) = (ID_A, ID_B) \vee (ID_B, ID_A)$. If the equation holds, $\mathcal{B}$ stops. Otherwise, $\mathcal{B}$ calculates the designated verifier's private key $\mathcal{S}_{j} = r_{j}cP$ for $\mathcal{F}$ to verify the exactness of signature $(\delta^{'}, \sigma^{'})$.

Step5. Finally, $\mathcal{F}$ can output the correct signature $(\delta^{'}, \sigma^{'}, m_{i})$, which is signed by ID_{i} and verified by the designated verifier ID_{j} and intended to be verified successfully using ID_A and ID_B, with non-negligible probability ב. If $\{ID_{i}, ID_{j}\} \neq \{ID_A, ID_B\} = \{aP, bP\}$, $\mathcal{B}$ outputs "failure" and aborts. Otherwise, $(ID_{i}, ID_{j}) = (ID_A, ID_B) \vee (ID_B, ID_A)$ holds, $\mathcal{F}$ will output $(\delta^{'}, \sigma^{'}, m_{i})$ with probability ב$/q(q-1)$.

$$\sigma^{'} = e(\delta^{'}, S_{j})^{H_2(m_{i}, \varepsilon^{'})+1}$$

$$\left(e(\xi^{'} + \mathcal{S}_{i}, Q_{j})^{\alpha^{'}}\right)^{\cdot \alpha^{'-1}} = \left(e(\delta^{'}, S_{j})^{H_2(m_{i}, \varepsilon^{'})+1}\right)^{\cdot \alpha^{'-1}}$$

$$= e(\delta^{'}, S_{j})^{H_2(m_{i}, \varepsilon^{'})} e(\delta^{'}, S_{j})^{\alpha^{'-1}}$$

$$\frac{e(\xi^{'} + \mathcal{S}_{i}, Q_{j})}{e(\delta^{'}, S_{j})^{H_2(m_{i}, \varepsilon^{'})}} = e(\delta^{'}, S_{j})^{\alpha^{'-1}} = e(\alpha^{'} Q_A, S_B)^{\alpha^{'-1}}$$

$$= e(Q_A, S_B)^{\alpha^{'-1} \cdot \alpha^{'}} = e(aP, cbP) = e(P, P)^{abc}$$

In other words, given (P, aP, bP, cP), $\mathcal{B}$ is able to compute $e(P, P)^{abc}$. That is $\mathcal{B}$ can break the BDH problem with non-negligible probability $\beth/q(q-1)$. But it is in contradiction with BDH assumption. ■

5.3. Source hiding

In our scheme, in the simulation stage, we don't use the identity of the signer. Hence, the verifier and any other party cannot know who was the signer. Even if an attacker can successfully intercept the transmitted signature (δ, σ), he can't know the signer's identity for the signature doesn't reveal any information about the signer's identity since it is protected by ECDLP. Therefore, our scheme can really hide the signer's identity.

5.4. Non-interactive

In our scheme, the designated verifier Bob uses only his secret key S_B in verifying the validity of the signature without the signer's cooperation. Hence, our scheme is non-interactive.

5.5. Deniable

In our scheme, the designated verifier could produce a signature to pass the verification equation. This makes the third party unable to distinguish who was the original signer. For example, $\sigma = e(\xi + S_A, Q_B)^{\alpha} = e(H_2(m, \varepsilon)S_A + S_A, Q_B)^{\alpha}$ $= e(\alpha Q_A + \alpha Q_A, S_B)^{H_2(m,\varepsilon)} = e(\tilde{\delta}, S_B)^{H_2(m,\varepsilon)+1}$. Bob can produce the same signature as Alice's. Hence, the signer can deny a signature signed by him.

5.6. Resistance against Conditional Key Compromise Impersonation (KCI) attack

Assume that two parties want to communicate with each other through Internet. KCI attack means that an attacker *E* knows the private key of A (B); he can masquerade as B (A) to communicate with A [18]. Now, we define conditional KCI attack as: *E* can pretend A to communicate with B, if he has B's private key, but B can't know A's identity.

Suppose that our scheme is applied to an electronic voting system, even the private key of the designated verifier is compromised, *E* can't masquerade as anyone to sign on a message to be verified successfully by the verifier since our scheme has the source hiding property. For example, in an open electronic voting system, each voter must be anonymous. Assume that an attacker *E* wants to masquerade as *C* to sign on a message *m* masquerade as *C* to vote a ballet to *V*. Even though he can know the private key of the verifier and can forge a signature on behalf of *C*. Since our

$\mathcal{F}$ $\mathcal{B}$

query to H_1 in $\mathcal{B}$ with (ID_i, Q_i) →

input (ID_i, Q_i)

output $\begin{cases} Q_i = aP, \text{if } ID_i = ID_A \\ Q_i = bP, \text{if } ID_i = ID_B \\ Q_i = r_iP, \text{if } ID_i \neq ID_A \text{ or } ID_B \end{cases}$

If $ID_i \neq ID_A$ or ID_B

returns $Q_i = H_1(ID_i) = r_iP$, $\mathcal{S}_i = r_icP$ as ID_i 's public/private key pair

← $(ID_i, Q_i, \mathcal{S}_i)$

inserts in H_1^{list}

query to H_2 in $\mathcal{B}$ with $P_i \in Z_q^*$ →

input (m_i, ε_i)

output P_i , where $P_i = H_2(m_i, \varepsilon_i)$ if (m_i, ε_i) exists in H_2-list

else P_i is a random chosen point in G_1

← P_i

inserts in H_2^{list}

input $ID_i \neq ID_A$ and ID_B

$ID_j \neq ID_A$ and ID_B

output $\mathcal{S}_i = r_icP$ and Q_i

$\mathcal{S}_j = r_jcP$ and Q_j

signing and verifying

$(\mathcal{S}_i,\ Q_i$ and $\mathcal{S}_j, Q_j)$, where (δ', σ') is both ID_i and ID_j's signature on *m* with $\{ID_i, ID_j\} \neq \{ID_A, ID_B\}$ ←

(δ', σ') →

checks weather

$(ID_i, ID_j) = (ID_A, ID_B) \vee (ID_B, ID_A) = (aP, bP) \vee (bP, aP)$,

if so (δ', σ', m_i) is also a correct signature made by ID_A and ID_B

output $e(P, P)^{abc}$ by the verification equation (as mentioned in step 5 of Section 5.2)

Figure 6 Signature forgery model

scheme has the property of source hiding, the verifier can't know who the signer was.

6. Comparisons and Discussions

6.1 Efficiency comparison

In the following, we make comparison of our proposed scheme with Laguillaumie *et.al's* [6], Zhang *et.al's* [9], Kang *et.al's* [14], and Kang *et.al's* [15], based on the length of the signature and the required computational cost. Here, we omit the comparison with [14](b), the second protocol of [14], since it is a proxy signature scheme. As shown in Table 1.

6.2 Security Comparisons

In this section**,** we make comparisons among our scheme and the other protocols proposed recently on the aspects of security features in Table 2.

We found that Lal *et al.*'s scheme [7] is insecure because it cannot resist against the forgery attack. Since the proxy signer "B" suffers from the attack that an adversary can masquerade as B to sign on a message which will be verified successfully by the two designated verifiers. Obviously, the *E* can replace $S_{ID_P} = rV + S_{ID_B}H_1(m_w)$ with $S_{ID_P} = r'V + S_{ID_E}H_1(m_w)$ which also will be verified successfully by the two designated verifiers, where $r' \in Z_q^*$ is a random number chosen by the adversary and $\sigma = (m_W, V)$ is the transmitted signature in the protocol, since *E* can produce α' to be verified successfully. In [14], we have demonstrated its weakness in Section 3. It suffers from the insider attack. In the aspect of Conditional KCI attack, all of the reviewed schemes [6, 9, 14(a), and 15] have not the property of source hiding. Because of the signer' public key was known to the verifier in the solution stage, this would enable an adversary to masquerade as the signer for communicating with the other verifier successfully in a multi-verifier scenario. Or once, the signer's identity recorded list has been stolen by a party, the party also can masquerade as the signer for communicating with the verifier.

In a word, our proposed scheme not only can prevent the attacks of insider, forgery, and conditional KCI but also possess the really source hiding which is a very important security feature needed in an electronic voting system.

6.3 Why our scheme really possesses the source hiding property?

After analysis, we found that schemes [1-6, 8-17] don't have the source hiding property despite the fact that among them schemes [1, 8, 9, 10, 12, 13, 15, 16] have claimed that they possess this property.

Table 1. Efficiency features comparisons

	Laguillaumie *et.al's* [6]	Zhang *et.al's* [9]	Kang *et.al's* [14](a)	Kang *et.al's* [15]	Our proposed scheme
Length	$\|G_1\|$	$\|G_1\|$	$2\|G_1\|$	$2\|G_1\|$	$\|G_1\|$
Pairing	2	2	2	3	3
multiplication	2	2	1	3	3
Exponentiation	0	0	0	2	2
Hash	2	2	2	2	1
Inverse	2	1	0	0	0

Table 2. Security features comparisons

protocols / properties	Laguillaumie *et al.'s* [6]	Lal *et al.*'s [7]	Zhang *et al.'s* [9]	Kang *et al.'s* [14](a)	Kang *et al.'s* [15]	Our proposed scheme
Insider attack prevention	✓	✓	✓	✓	✓	✓
Forgery attack prevention	✓		✓	✓	✓	✓
Conditional KCI attack prevention						✓
Source hiding		✓				✓

This is because their schemes incorporate the signer's public key into the verification and simulation phases. Conversely, in our scheme, a verifier needs not be aware of the signer's public key in the verification and simulation phases. Hence, our protocol really has the source hiding property.

7. Conclusion

In this paper, we show that all of the proposed DVS [1-17], expert for the proxy signature [7, 14(b)], haven't the source hiding property. Besides, we have proposed a provably secure and source hiding DVS scheme which can resist against all known attacks we have shown its security based on the random oracle model.

After comparisons, we conclude that our scheme not only is the most secure but also is the only scheme that possesses source hiding property. This makes our scheme be suitable for the application in an election voting system. Because in an election voting system, the tally can’t know who is the voter. In other words, the tally can’t know who the original signer on the vote was.

Reference

[1] M. Jakobsson, K. Sako, R. Impagliazzo, “Designated verifier proofs and their applications, ”Advances in Cryptology — EUROCRYPT '96, *Lecture Notes in Computer Science*, vol. 1070, Springer, Berlin, (1996), pp. 143–154.

[2] G. Wang “An Attack on Not-interactive Designated Verifier Proofs for Undeniable Signatures, ”*Cryptology ePrint Archive*: Report 2003/243

[3] F. Laguillaumie and D. Vergnaud “Multi-Designated Verifiers Signatures, ”*Lecture Notes in Computer Science*, (2004) – Springer

[4] F. Laguillaumie and D. Vergnaud “Designated Verifier Signatures: Anonymity and Efficient Construction from any Bilinear Map, ”*Proc. of SCN*, (2004) – Springer

[5] S. Lal and V. Verma “Identity Based Strong Designated Verifier Proxy Signature Schemes, ”*Cryptology ePrint Archive*: Report 2006/394

[6] F. Laguillaumie, D. Vergnaud “Multi-designated verifiers signatures: anonymity without encryption, ”*Information Processing Letters* 102 (2007) 127–132

[7] S Lal and V Verma “Identity Based Strong Bi-Designated Verifier Proxy Signature Schemes, ”*Cryptology ePrint Archive*: Report 2008/024

[8] H. Du and Q. Wen “Attack on Kang et al’s Identity-Based Strong Designated Verifier Signature Scheme, ”*Cryptology ePrint Archive*: Report 2008/297

[9] J. Zhang and J. Mao “A novel ID-based designated verifier signature scheme, ”*Information Sciences*, Volume 178, Issue 3, 1 February (2008), Pages 766-773

[10] S.H. Seo, J.Y. Hwang, K.Y. Choi and D.H. Lee “Identity-based universal designated multi-verifiers signature schemes, ”*Computer Standards & Interfaces*, Volume 30, Issue 5, July (2008), Pages 288-295

[11]K.A. Shim “Rogue-key attacks on the multi-designated verifiers signature scheme, ”*Information Processing Letters* 107 (2008) 83–86

[12]B. Wang, Z.X. Song “A non-interactive deniable authentication scheme based on designated verifier proofs, ”*Information Sciences*, Volume 179, Issue 6, 1 March (2009), Pages 858-865

[13]J.S. Lee, J.H. Chang “Comment on Saeednia et al.'s strong designated verifier

signature scheme, "*Computer Standards & Interfaces*, January (2009), Pages 258-260

[14] B. Kang, C. Boyd and E. Dawson "Identity-based strong designated verifier signature schemes: Attacks and new construction, "*Computers & Electrical Engineering*, Volume 35, Issue 1, January (2009), Pages 49-53

[15] B. Kang, C. Boyd and Ed Dawson "A novel identity-based strong designated verifier signature scheme, "*Journal of Systems and Software*, Volume 82, Issue 2, February (2009), Pages 270-273

[16] Y. Yu, C. Xu, X. Zhang and Y. Liao "Designated verifier proxy signature scheme without random oracles, "*Computers & Mathematics with Applications*, Volume 57, Issue 8, April (2009), Pages 1352-1364

[17] F. Cao and Z. Cao "An identity based universal designated verifier signature scheme secure in the standard model, "*Journal of Systems and Software*, Volume 82, Issue 4, April (2009), Pages 643-649

[18] S. B. Wilson, and A. Menezes, "Authenticated Diffie-Hellman key agreement protocols, " *Lecture Notes in Computer Science*, 1999 - Springer