JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 451-463 (2011)

A DL Based Short Strong Designated Verifier Signature Scheme with Low Computation

HAN-YU LIN, TZONG-SUN WU+ AND YI-SHIUNG YEH
Department of Computer Science
National Chiao Tung University
Hsinchu, 300 Taiwan
+*Department of Computer Science and Engineering*
National Taiwan Ocean University
Keelung, 202 Taiwan

A designated verifier signature (DVS) scheme is a special type of digital signature scheme without the property of non-repudiation. Such schemes only allow the designated verifier to validate the signer's signature. Meanwhile, the designated verifier is not able to convince any third party of the signature's actual signer, since he can also generate a computationally indistinguishable one compared to the received signature. A strong designated verifier signature (SDVS) scheme further prohibits anyone except for the intended verifier from validating the signature as it requires the designated verifier's private key to finish the verification process. In this paper, we propose an efficient short SDVS scheme with low computation costs. Unlike many SDVS schemes primarily implemented on pairing-based cryptosystems, we focus on the discrete logarithms (DL) based system. Compared with those DL based SDVS schemes, ours provides better functionalities and efficiency and hence benefits the practical applications. In addition, the security requirement of unforgeability against existential forgery on adaptive chosen-message attacks (EU-CMA) is proved in the random oracle model.

Keywords: designated verifier, digital signature, discrete logarithms, public key system, cryptography

1. INTRODUCTION

A digital signature scheme [1, 2] is an important technique of public key cryptosystems [2-5] to ensure the properties of integrity, authenticity [6] and non-repudiation [7]. Since all public keys are either maintained by the system authority (SA) or stored in the public key directory, one can easily obtain the corresponding public key of the other to verify his/her signature. The actual signer thus can not deny his/her generated signature later. However, in some applications such as the electronic voting [8, 9], the non-repudiation property is not desirable.

With an eye to the above requirement, in 1990, Chaum and Antwerpen [10] proposed an undeniable signature scheme in which the signer must assist the verifier to validate the resulted signature. It is obviously that any third party attempting to verify the signature has to reach an agreement with the signer in advance. That is to say, in an undeniable signature scheme, the signer has completely control over his generated signatures. In 1996, Jakobsson *et al*. [11] came up with the notion of designated verifier proofs and in a sense proposed a designated verifier signature (DVS) scheme. In their scheme, the designated

Received June 15, 2009; revised November 11 & December 22, 2009; accepted February 9, 2010.
Communicated by Chin-Laung Lei.
+ Corresponding author.

verifier can be convinced of the signer's identity regarding a given signature without the assistance of the actual signer. Yet, the designated verifier can not transfer the proofs to convince any third party, since he is also capable of generating another DVS which is computationally indistinguishable from the received one. In 2003, Wang [12] pointed out that Jakobsson *et al.*'s scheme is insure because a malicious signer can easily cheat the designated verifier. In the same year, Saeednia *et al.* [13] formalized the notion of DVS scheme and further proposed a so-called strong designated verifier signature (SDVS) scheme in which the designated verifier's private key is directly involved in the validation equation. Consequently, anyone can not even perform the validation equation without the knowledge of designated verifier's private key. In 2007, Lee and Chang [14] further combine SDVS schemes with message recovery signatures. More recently, they [15] pointed out that signer's ambiguity could be a vital property of secure SDVS schemes. Namely, even if a signer's private key is compromised, any attacker still can not identify the actual signer for a given SDVS which has not been received by the designated verifier. Another SDVS scheme satisfying such a property is also proposed in their scheme. Nevertheless, they give no formal proof. In 2004, Susilo *et al.* [16] addressed the first identity-based SDVS scheme from bilinear pairings. Since then, several researchers [17-20] have devoted themselves to the design of pairings-based SDVS schemes. However, we find out that none of these schemes could fulfill the property of signer's ambiguity addressed by Lee and Chang [15].

Generally speaking, an SDVS scheme should satisfy the following security requirements:

(1) **Unforgeability:** It is computationally infeasible for any polynomial-time adversary to forge a valid SDVS without knowing the private key of either the signer or the designated verifier.
(2) **Non-Transferability:** Based on the transcript simulation property in an SDVS scheme, the designated verifier can also generate another SDVS which is computationally indistinguishable from the received one. Therefore, the designated verifier can not transfer the SDVS to any third party.
(3) **Signer's Ambiguity:** It is difficult to determine the identity of signer from the actual signer and the designated verifier for a given SDVS.

In this paper, we focus on the DL based systems and propose a provably secure short SDVS scheme with low computation costs in the random oracle model. Compared with related works, our scheme not only has shorter signature length, but also earns more computational efficiency.

The rest of this paper is organized as follows. Section 2 states some preliminaries. We introduce the proposed short SDVS scheme in section 3. The security proof and some comparisons are detailed in section 4. Finally, a conclusion is made in section 5.

2. PRELIMINARIES

In this section, we briefly review some security notions and the computational assumptions.

Discrete Logarithm Problem; DLP [21]

Let p and q be two large primes satisfying $q|p-1$, and g a generator of order q over GF(p). The discrete logarithm problem is, given an instance (y, p, q, g), where $y = g^x \bmod p$ for some $x \in Z_q$, to derive x.

Discrete Logarithm (DL) Assumption [22]

A probabilistic polynomial-time algorithm $\mathcal{B}$ is said to (t, ε)-break the DLP if given a DLP instance (y, p, q, g) where $y = g^x \bmod p$ for some $x \in Z_q$, $\mathcal{B}$ can derive x with probability ε after running at most t steps. The probability is taken over the uniformly and independently chosen instance and over the random bits consumed by $\mathcal{B}$.

Definition 1 The (t, ε)-DL assumption holds if there exists no probabilistic polynomial-time adversary that can (t, ε)-break the DLP.

3. THE PROPOSED SCHEME

In this section, we first addresses involved parties and composed algorithms of our proposed scheme and then give a concrete construction.

3.1 Involved Parties

An SDVS scheme has two involved parties: a signer and a designated verifier. Each one is a probabilistic polynomial-time Turing machine (PPTM). The signer will generate an SDVS intended for the designated verifier. Consequently, the resulted SDVS can only be validated by the designated verifier with his private key. An SDVS scheme is correct if the signer can generate a valid SDVS and only the designated verifier can be convinced of the signer's identity.

3.2 Algorithms

The proposed SDVS scheme consists of the following algorithms:

Setup: Taking as input 1^k where k is a security parameter, the algorithm generates the system's public parameters *params*.

Signature-Generation (SG): The SG algorithm takes as input the system parameters *params*, a message, the public key of the target designated verifier and the private key of signer. It generates the resulted SDVS δ.

Signature-Verification (SV): The SV algorithm takes as input the system parameters *params*, a message m, an SDVS δ, the private key of the designated verifier and the public key of signer. It outputs **True** if the δ is a valid SDVS for m. Otherwise, the symbol $\perp$ is returned as a result.

Transcript-Simulation (TS): The TS algorithm takes as input the system parameters *params*, a message m, an SDVS δ and the private key of designated verifier. It outputs another valid SDVS δ^* for m.

3.3 Concrete Construction

We demonstrate the proposed short SDVS scheme over a finite field. Details are described below:

Setup: Taking as input 1^k, the system authority (SA) selects two large primes p and q where $|q| = k$ and $q \mid (p - 1)$. Let g be a generator of order q and f: $Z_p^* \times Z_p^* \rightarrow Z_q$, F: $Z_q \rightarrow Z_q$ and H: $\{0, 1\}^* \times Z_q \rightarrow Z_q$ collision resistant hash functions. The system publishes the public parameters $params = \{p, q, g, f, F, H\}$. Each user U_i chooses his private key $x_i \in Z_q$ and computes the public key as $y_i = g^{x_i} \bmod p$. In addition, he also announces a universal parameter $T_i = g^{c_i} \bmod p$ where $c_i \in_R Z_q$.

Signature-Generation (SG): Let U_s and U_v be a signer and a designated verifier, respectively. For signing a message $m \in_R \{0, 1\}^*$, U_s first chooses $w \in_R Z_q$ to compute $Q = F(w)$ and

$$R = f(y_v^w \bmod p, y_v^{c_s} \bmod p), \tag{1}$$
$$S = (w - x_s H(m, Q, T_s)) \bmod q. \tag{2}$$

Then U_s deliveries m along with its SDVS $\delta = (Q, R, S)$ to U_v.

Signature-Verification (SV): Upon receiving (m, δ), U_v computes

$$Z_1 = y_v^S y_s^{x_v H(m,Q,T_s)} \bmod p, \tag{3}$$
$$Z_2 = T_s^{x_v} \bmod p, \tag{4}$$

and then verifies the signature by checking whether

$$R = f(Z_1, Z_2). \tag{5}$$

We show that the verification of Eq. (5) works correctly. From the right-hand side of Eq. (5), we have

$$\begin{aligned} f(Z_1, Z_2) &= f(y_v^S y_s^{x_v H(m,Q,T_s)} \bmod p, T_s^{x_v} \bmod p) && \text{(by Eqs. (3) and (4))} \\ &= f(y_v^{S + x_s(H(m,Q,T_s)} \bmod p, T_s^{x_v} \bmod p) \\ &= f(y_v^{S + x_s(H(m,Q,T_s)} \bmod p, y_v^{c_s} \bmod p) \\ &= f(y_v^w \bmod p, y_v^{c_s} \bmod p) && \text{(by Eq. (2))} \\ &= R && \text{(by Eq. (1))} \end{aligned}$$

which leads to the left-hand side of Eq. (5).

Transcript-Simulation (TS): To generate another SDVS δ^* intended for himself, U_v computes

$$S^* = S + 1 \bmod q, \tag{6}$$
$$R^* = f(y_v Z_1 \bmod p, Z_2). \tag{7}$$

Here, $\delta^* = (Q, R^*, S^*)$ is another valid SDVS for the message m. In fact, the probability

that the computed $\delta^* = (Q, R^*, S^*)$ and the received $\delta = (Q, R, S)$ are identical is at most $1/2^k$, *i.e.*, $\Pr[\delta^* = \delta] \leq 1/2^k$.

4. SECURITY PROOF AND COMPARISON

In this section, we first define the security model of our proposed SDVS scheme and prove it in the random oracle model. Then some comparisons with related schemes are made.

4.1 Security Model

The crucial security requirement of the proposed SDVS scheme is unforgeability against existential forgery on adaptive chosen-message attacks (EU-CMA). We define the notion as follows,

Definition 2 The proposed SDVS scheme is said to $(t, q_F, q_H, q_{SG}, q_{SV}, \varepsilon)$-secure against adaptive chosen message attacks (EU-CMA) if there exists no probabilistic polynomial-time adversary $\mathcal{A}$ running at most t steps, asking at most q_F F random oracle, q_H H random oracle, q_{SG} SG and q_{SV} SV queries and then winning the following game with non-negligible advantage ε.

Setup: A challenger $\mathcal{B}$ first runs the Setup(1^k) algorithm and sends the system's public parameters *params* to the adversary $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ can make several kinds of queries adaptively, *i.e.*, each query might be based on the result of previous queries:

- F random oracle queries: For each F random oracle query asked by $\mathcal{A}$, $\mathcal{B}$ responses with a random number and maintains a table to store it.
- H random oracle queries: For each H random oracle query asked by $\mathcal{A}$, $\mathcal{B}$ responses with a random number and maintains a table to store it.
- Signature-Generation (SG) queries: $\mathcal{A}$ makes an SG query for a message m with respect to the signer and the target designated verifier. $\mathcal{B}$ runs the SG algorithm on behalf of the signer and returns the corresponding SDVS.
- Signature-Verification (SV) queries: $\mathcal{A}$ gives $\mathcal{B}$ a pair (m, δ). $\mathcal{B}$ runs the SV algorithm on behalf of the designated verifier and outputs **True** to $\mathcal{A}$ if δ is a valid SDVS for m. Otherwise, the symbol $\perp$ is returned as a result.

Forgery: Finally, $\mathcal{A}$ produces a new pair (m^*, δ^*) which is not outputted by the SG query. The adversary $\mathcal{A}$ wins if δ^* is a valid SDVS for m^*.

4.2 Security Proof

We prove that the proposed scheme achieves the EU-CMA security in the random oracle model. As our scheme is motivated by Schnorr's signature scheme [23] which can

be regarded as a generic signature scheme, we can directly apply the proof techniques addressed by Pointcheval and Stern [24] to obtain the following theorem.

Theorem 1 (The Forking Lemma) In the random oracle mode, let $\mathcal{A}$ be a probabilistic polynomial-time Turing machine whose input only consists of public data. We denote respectively by N_1 and N_2 the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the signer. Assume that, within a time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geq 10(N_2 + 1)(N_2 + N_1)/2^k$, a valid signature $(m, \sigma_1, h, \sigma_2)$ where $\sigma_1 = R$, $h = (H(m, Q, T_s), F(w))$ and $\sigma_2 = S$. If the triples (σ_1, h, σ_2) can be simulated without knowing the private key with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from $\mathcal{A}$ replacing interaction with the signer by simulation and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $H(m, Q, T_s) \neq H'(m, Q, T_s)$ in the expected time $T' \leq 120686T/\varepsilon$.

Concretely speaking, in our scheme, we can first obtain two equations

$$Z_1 = y_v^{S} y_s^{x_v H(m,Q,T_s)} \bmod p,$$
$$Z_1 = y_v^{S'} y_s^{x_v H'(m,Q,T_s)} \bmod p,$$

and then compute the private key x_s as

$$x_s = (S - S')/(H'(m, Q, T_s) - H(m, Q, T_s)).$$

Still, to show the tight relation between the security of our SDVS scheme and the hardness of the DLP, we have to present another more detailed security proof and advantage analyses as follows.

Theorem 2 The proposed SDVS scheme is $(t, q_F, q_H, q_{SG}, q_{SV}, \varepsilon)$-secure against existential forgery on adaptive chosen-message attacks (EU-CMA) in the random oracle model if there exists no probabilistic polynomial-time adversary that can (t', ε')-break the DLP, where

$$\varepsilon' \geq (q_F^{-1})(\varepsilon - 2^{-k}) + ((q_F - 1)q_F^{-1})(4^{-1}(\varepsilon - 2^{-k})^3(q_F^{-1} + q_H^{-1})),$$
$$t' \approx t + t_\lambda(2q_{SG} + 2q_{SV}).$$

Here t_λ is the costs for performing a modular exponentiation over a finite field.

Proof: Fig. 1 depicts the proof structure of this theorem. Suppose that a probabilistic polynomial-time adversary $\mathcal{A}$ can $(t, q_F, q_H, q_{SG}, q_{SV}, \varepsilon)$-break the proposed SDVS scheme with non-negligible advantage ε under adaptive chosen-message attacks after running at most t steps and asking at most q_F F random oracle, q_H H random oracle, q_{SG} SG and q_{SV} SV queries. Then we can construct another algorithm $\mathcal{B}$ that can (t', ε')-break the DLP by taking $\mathcal{A}$ as a subroutine. Let all involved parties and notations be defined the same as those in section 3.3, f a collision resistant hash function and (H, F) random oracles. The objective of $\mathcal{B}$ is to obtain α $(= \log_g C)$ by taking $(p, q, g, C = g^\alpha \bmod p)$ as inputs. In this proof, $\mathcal{B}$ simulates a challenger to $\mathcal{A}$ in the following game.

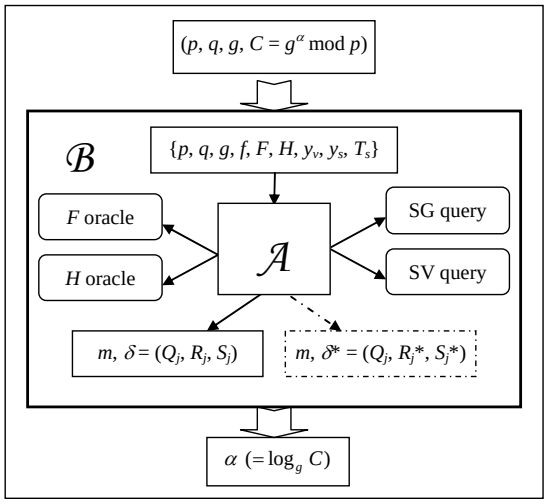

Fig. 1. The proof structure of Theorem 2.

Setup: The challenger $\mathcal{B}$ runs the Setup(1^k) algorithm to obtain the system's public parameters *params* = {*p*, *q*, *g*, *f*, *F*, *H*} and comes up with a random tape composed of a long sequence of random bits. Then $\mathcal{B}$ chooses r, $c_s \in_R Z_q$ to set $y_v = g^r \bmod p$, $y_s = C$ and $T_s = g^{c_s} \bmod p$. After that, $\mathcal{B}$ simulates one or two runs of SDVS scheme to the adversary $\mathcal{A}$ on input {*p*, *q*, *g*, *f*, *F*, *H*, y_v, y_s, T_s} and the random tape.

Phase 1: $\mathcal{A}$ makes the following kinds of queries adaptively:

- *F* random oracle: When $\mathcal{A}$ queries an *F* oracle of *F*(*w*), $\mathcal{B}$ returns ***O*-Sim_*F***(*w*). The simulated random oracle ***O*-Sim_*F*** operates as Fig. 2. Note that the function **insert**(*N*, *b*) will insert the value *b* into the array *N*.
- *H* random oracle: When $\mathcal{A}$ queries an *H* oracle of *H*(*m*, *Q*, T_s), $\mathcal{B}$ returns ***O*-Sim_*H***(*m*, *Q*, T_s). The simulated random oracle ***O*-Sim_*H*** operates as Fig. 3.
- SG queries: When $\mathcal{A}$ makes an SG query for some message *m*, $\mathcal{B}$ returns (*m*, ***O*-Sim_*SG***(*m*)) as the result. The simulated SG oracle ***O*-Sim_*SG*** operates as Fig. 4.
- SV queries: When $\mathcal{A}$ makes an SV query for some pair (*m*, δ), $\mathcal{B}$ returns ***O*-Sim_*SV***(*m*, δ) as the result. The simulated SV oracle ***O*-Sim_*SV*** operates as Fig. 5.

```
oracle O-Sim_F(w)
1: for i = 0 to q_F − 1
2:     if (Q_F[i] = w) then      // It is an old query.
3:         exit for;
4:     else if (Q_F[i] = "") then      // It is a new query.
5:         insert(Q_F, w); A_F[i] ← l ∈_R Z_q; exit for;
6:     end if
7: next i
8: return A_F[i];
```

Fig. 2. Algorithm of the simulated random oracle ***O*-Sim_*F***.

oracle *O*-Sim_*H*(*m*, *Q*, T_s)
1: for $i = 0$ to $q_H - 1$
2: if ($Q\_H[i][0] = m$ and $Q\_H[i][1] = Q$ and $Q\_H[i][2] = T_s$) then // It is an old query.
3: exit for;
4: else if ($Q\_H[i] =$ "") then // It is a new query.
5: **insert**(Q_H, (m, Q, T_s)); $A\_H[i] \leftarrow h \in_R Z_q$; exit for;
6: end if
7: next i
8: return $A\_H[i]$;

Fig. 3. Algorithm of the simulated random oracle ***O*-Sim_*H***.

oracle *O*-Sim_*SG*(*m*)
1: Choose w, $S \in_R Z_q$;
2: $Q \leftarrow$ ***O*-Sim_*F*(*w*)**; $h \leftarrow$ ***O*-Sim_*H*(*m*, *Q*, T_s)**;
3: $Z_1 = y_v^S y_s^{rh} \bmod p$; $Z_2 = y_v^{c_s} \bmod p$; $R = f(Z_1, Z_2)$;
4: return $\delta = (Q, R, S)$;

Fig. 4. Algorithm of the simulated SG oracle ***O*-Sim_*SG***.

oracle *O*-Sim_*SV*(*m*, δ) // $\delta = (Q, R, S)$
1: $h \leftarrow$ ***O*-Sim_*H*(*m*, *Q*, T_s)**;
2: $Z_1 = y_v^S y_s^{rh} \bmod p$; $Z_2 = T_s^r \bmod p$; $R^* = f(Z_1, Z_2)$;
3: if ($R = R^*$) then
4: return **True**;
5: else
6: return $\perp$;
7: end if

Fig. 5. Algorithm of the simulated SV oracle ***O*-Sim_*SV***.

Analysis of the game: It can be seen that the simulation is almost perfect and the adversary $\mathcal{A}$'s view in the above simulation is computationally indistinguishable from that in a real situation. Let Fv be the event that $\mathcal{A}$ tries to forge an SDVS for a message m and then finally outputs a valid SDVS $\delta = (Q_j, R_j, S_j)$. By assumption, we know that $\mathcal{A}$ has non-negligible probability ε to break the proposed SDVS scheme, *i.e.*, Pr[Fv] = ε. The probability that $\mathcal{A}$ guesses a correct random oracle value without asking $F(w_j)$ or $H(m, Q_j, T_s)$ is not greater than 2^{-k}. We denote such an event by NRO and Pr[NRO] $\leq 2^{-k}$. Consequently, we can further express the probability that $\mathcal{A}$ outputs a valid forgery after asking $F(w_j)$ along with $H(m, Q_j, T_s)$ queries as

$$\Pr[\text{Fv} \wedge \neg\text{NRO}] \geq (\varepsilon - 2^{-k}).$$

If $\mathcal{A}$ has ever asked $F(w_j)$, w_j must be kept in some entry of the Q_F array. Then $\mathcal{B}$ will have the probability of q_F^{-1} to solve the DLP by computing

$$\alpha = (w_j - S_j)H(m, Q_j, T_s)^{-1} \bmod q.$$

In the other hand, with the probability of $(q_F - 1)q_F^{-1}$, $\mathcal{B}$ might have to launch the second simulation in case that $\mathcal{A}$ didn't ask $F(w)$. $\mathcal{B}$ again runs $\mathcal{A}$ on input {p, q, g, f, F, H,

y_v, y_s, T_s} and the same random tape. Since $\mathcal{A}$ is provided with the same sequence of random bits, we know that the ith query he will ask is always the same as the one during the first simulation. For all the oracle queries before $H(m, Q_j, T_s)$, $\mathcal{B}$ returns identical results as those in the first time. When $\mathcal{A}$ asks $H(m, Q_j, T_s)$, $\mathcal{B}$ directly gives a new $h_j^* \in_R Z_q$ instead of h_j. Meanwhile, $\mathcal{A}$ is then provided with another different random tape which is also composed of a long sequence of random bits. By the "Forking lemma" of Theorem 1, if $\mathcal{A}$ eventually outputs another valid SDVS $\delta^* = (Q_j, R_j^*, S_j^*)$ with $H(m, Q_j, T_s) \neq H^*(m, Q_j, T_s)$ or $\mathcal{A}$ has ever asked $F(w_j)$ this time, $\mathcal{B}$ would have a chance to solve the DLP. To evaluate $\mathcal{B}$'s success probability, we use the "Splitting lemma" [24] as follows.

Let X and Y be the sets of possible sequences of random bits and random function values supplied to $\mathcal{A}$ before and after the $H(m, Q_j, T_s)$ query is made by $\mathcal{A}$, respectively. Then we know that on inputting a random value $(x \| y)$ for any $x \in X$ and $y \in Y$, $\mathcal{A}$ outputs a valid forgery with the probability of ε, *i.e.*, $\Pr_{x \in X, y \in Y}[\mathrm{Fv}] = \varepsilon$. According to the "Splitting lemma", there exists a subset $D \in X$ such that

(a) $\Pr[x \in D] = |D| \cdot |X|^{-1} \geq 2^{-1}\varepsilon$.
(b) $\forall x \in D$, $\Pr_{y \in Y}[\mathrm{Fv}] \geq 2^{-1}\varepsilon$.

From the above definition, we know that if $n \in D$ is the supplied sequence of random bits and random function values given to $\mathcal{A}$ before the $H(m, Q_j, T_s)$ query is made, then for any sequence of random bits and random function values $y' \in Y$ after the query, $\mathcal{A}$ outputs a valid forgery with the probability of at least $(2^{-1}\varepsilon)^2 = 4^{-1}\varepsilon^2$, *i.e.*,

$$\Pr_{n \in D, y' \in Y}[\mathrm{Fv}] \geq 4^{-1}\varepsilon^2.$$

Since the probability that $\mathcal{A}$ outputs another SDVS $\delta^* = (Q_j, R_j^*, S_j^*)$ with $H(m, Q_j, T_s) \neq H^*(m, Q_j, T_s)$ is q_H^{-1}, we can express the probability that $\mathcal{B}$ solve the DLP in the second simulation as

$$(\varepsilon - 2^{-k})(4^{-1}(\varepsilon - 2^{-k})^2)(q_F^{-1} + q_H^{-1}) = 4^{-1}(\varepsilon - 2^{-k})^3(q_F^{-1} + q_H^{-1}).$$

By adding the success probability of $\mathcal{B}$ in the first simulation, one can observe that after the second simulation, $\mathcal{B}$ could solve the DLP with the success probability

$$\varepsilon' \geq (q_F^{-1})(\varepsilon - 2^{-k}) + ((q_F - 1)q_F^{-1})(4^{-1}(\varepsilon - 2^{-k})^3(q_F^{-1} + q_H^{-1})).$$

Moreover, the computational steps required for $\mathcal{B}$ during one simulation is

$$t + t_\lambda(2q_{SG} + 2q_{SV})$$

where t_λ is the costs for performing a modular exponentiation over a finite field. We therefore can represent the total computational steps for $\mathcal{B}$ after the second simulation as

$$t' \approx (q_F^{-1})(t + t_\lambda(2q_{SG} + 2q_{SV})) + ((q_F - 1)q_F^{-1})(t + t_\lambda(2q_{SG} + 2q_{SV}))$$
$$= t + t_\lambda(2q_{SG} + 2q_{SV}).$$ ❑

4.3 Comparisons

For facilitating the following comparisons, we first define several used notations:

$|x|$: the bit-length of an integer x
H: a one-way hash function
M: a modular multiplication computation
E: a modular exponentiation computation
I: a modular inverse computation

The time for performing the modular addition and subtraction computation is ignored because it is negligible as compared to the above. We compare our scheme with previously proposed DL ones including Jakobsson *et al.*'s (JSI for short) [11], Saeednia *et al.*'s (SKM for short) [13], the Yang-Liao (YL for short) [25] and two presented by Lee and Chang separately in 2007 (LC-1 for short) [14] and 2009 (LC-2 for short) [15]. Detailed comparisons are demonstrated as Table 1. Although the Yang-Liao scheme has the lowest computational costs, the signature length of their scheme is longer than that of ours. Most importantly, their scheme cannot satisfy the requirement of signer's ambiguity addressed in [15], which is regarded as an essential property of secure SDVS schemes. To sum up, we conclude that the proposed SDVS scheme not only provides better functionalities, but also has lower computation costs and shorter signature length.

Table 1. Comparisons of the proposed and related schemes.

Item \ Scheme	JSI	SKM	YL	LC-1	LC-2	Ours
Basic Assumption	DL	DL	DL	DL	DL	DL
Unforgeability	×	√	√	√	√	√
Non-Transferability	√	√	√	√	√	√
Signer's Ambiguity	×	×	×	×	√	√
Provable Security	×	×	√	×	×	√
Signature Length[1]	$3\|p\| + 3\|q\|$ ≈ 2016 bits	$3\|q\|$ ≈ 480 bit	$\|p\| + \|q\|$ ≈ 672 bits	$2\|p\| + 2\|q\|$ ≈ 1344 bits	$\|p\| + \|q\|$ ≈ 672 bits	$3\|q\|$ ≈ 480 bits
Computation Costs[2]	$16E + 8M + 3H + I$	$6E + 8M + 3H + 4I$	$3E + 6M + 3H$	$12E + 10M + 3H + 2I$	$7E + 3M + 3H$	$5E + 4M + 6H$

1. Let $|p| \approx 512$ bits and $|q| \approx 160$ bits.
2. The computation costs include phases of signature generation, signature verification and transcript simulation.

5. CONCLUSIONS

In this paper, we have proposed an efficient DL based short SDVS scheme with low computation. Since the private key of designated verifier is directly involved in the signature validation process, only the designated verifier can be convinced of the signer's identity. Any third party including the signer can not check the signature validation equation without having the knowledge of the designated verifier's private key. Compared with previous related works, our proposed short SDVS scheme has better efficiency and func-

tionalities. Moreover, we also proved that the proposed SDVS scheme achieves the EU-CMA security in the random oracle model.

REFERENCES

1. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, Vol. IT-31, 1985, pp. 469-472.
2. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, Vol. 21, 1978, pp. 120-126.
3. W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. IT-22, 1976, pp. 644-654.
4. M. Girault, "Self-certified public keys," in *Advances in Cryptology – EUROCRYPT*, 1991, pp. 491-497.
5. A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology – CRYPTO*, 1984, pp. 47-53.
6. W. Stallings, *Cryptography and Network Security: Principles and Practices*, 4th ed., Pearson, New York, 2005.
7. B. Meng, S. Wang, and Q. Xiong, "A fair non-repudiation protocol," in *Proceedings of the 7th International Conference on Computer Supported Cooperative Work in Design*, 2002, pp. 68-73.
8. I. Ray and N. Narasimhamurthi, "An anonymous electronic voting protocol for voting over the Internet," in *Proceedings of the 3rd International Workshop on Advanced Issues of e-Commerce and Web-Based Information Systems*, 2001, pp. 188-190.
9. B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Advances in Cryptology – CRYPTO*, 1999, pp. 148-164.
10. D. Chaum and H. van Antwerpen, "Undeniable signature," in *Advances in Cryptology – CRYPTO*, 1990, pp. 212-216.
11. M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," in *Advances in Cryptology – EUROCRYPT*, 1996, pp. 143-154.
12. G. Wang, "An Attack on not-interactive designated verifier proofs for undeniable signatures," *Cryptology ePrint Archive*, http://eprint.iacr.org/2003/243, 2003.
13. S. Saeednia, S. Kremer, and O. Markowitch, "An efficient strong designated verifier signature scheme," in *Proceedings of the 6th International Conference on Information Security and Cryptology*, 2003, pp. 40-54.
14. J. S. Lee and J. H. Chang, "Strong designated verifier signature scheme with message recovery," in *Proceedings of the 9th International Conference on Advanced Communication Technology*, Vol. 1, 2007, pp. 801-803.
15. J. S. Lee and J. H. Chang, "Comment on Saeednia *et al.*'s strong designated verifier signature scheme," *Computer Standards and Interfaces*, Vol. 31, 2009, pp. 258-260.
16. W. Susilo, F. Zhang, and Y. Mu, "Identity-based strong designated verifier signature schemes," *Information Security and Privacy*, Vol. 3108, 2004, pp. 167-170.
17. X. Huang, W. Susilo, Y. Mu, and F. Zhang, "Short designated verifier signature scheme and its identity-based variant," *International Journal of Network Security*, Vol. 6, 2008, pp. 82-93.

18. B. Kang, C. Boyd, and E. Dawson, "A novel identity-based strong designated verifier signature scheme," *The Journal of Systems and Software*, Vol. 82, 2009, pp. 270-273.
19. K. Kumar, G. Shailaja, and A. Saxena, "Identity based strong designated verifier signature scheme," *Cryptology ePrint Archive*, Report 2006/134, http://eprint.iacr.org/2006/134, 2006.
20. J. Zhang and J. Mao, "A novel ID-based designated verifier signature scheme," *Information Sciences*, Vol. 178, 2008, pp. 766-773.
21. H. Delfs and H. Knebl, *Introduction to Cryptography: Principles and Applications*, Springer, Berlin, 2002.
22. Z. Shao, "A provably secure short signature scheme based on discrete logarithms," *Information Sciences*, Vol. 177, 2007, pp. 5432-5440.
23. C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, Vol. 4, 1991, pp. 161-174.
24. D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, Vol. 13, 2000, pp. 361-369.
25. F. Y. Yang and C. M. Liao, "A provably secure and efficient strong designated verifier signature scheme," *International Journal of Network Security*, Vol. 10, 2010, pp. 223-227.

Han-Yu Lin (林韓禹) received his B.A. degree in Economics from the Fu Jen Catholic University, Taiwan in 2001, and his M.S. degree in Information Management from the Huafan University, Taiwan in 2003. Now he is a Ph.D. candidate in the Department of Computer Science of National Chiao Tung University, Taiwan. His research interests include cryptology and network security.

Tzong-Sun Wu (吳宗杉) received his B.S. degree in Electrical Engineering from the National Taiwan University, Taiwan in 1990, and his Ph.D. in Information Management from the National Taiwan University of Science and Technology, Taiwan in 1998. From August 1998 to July 2002, he has been an Assistant Professor in the Department of Information Management of Huafan University. From August 2002 to January 2007, he has been an Associate Professor in the Department of Informatics of Fo Guang University. He is now with the Department of Computer Science and Engineering, National Taiwan Ocean University. His research interests include information security, watermarking, digital right management, and e-commerce.

Yi-Shiung Yeh (葉義雄) received his M.S. and Ph.D. degrees in Department of Electrical Engineering and Computer Science from University of Wisconsin-Milwaukee, U.S.A. in 1980 and 1986, respectively. He was a Professor in the Department of Computer Science, National Chiao Tung University, Taiwan till 2007. His research interests include data security and privacy, information and coding theory, game theory, reliability, and performance.