Toward Secure Strong Designated Verifier Signature Scheme from Identity-Based System

Han-Yu Lin
Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan

Abstract: *A Designated Verifier Signature (DVS) scheme has the property of signer ambiguity, which can only convince a designated verifier of the signer's identity. It is not allowed for the designated verifier to transfer the conviction to any third party. A Strong Designated Verifier Signature (SDVS) scheme further requires the designated verifier's private key to perform the signature verification, so as to prevent anyone else from validating the signature. However, once the signer's private key is compromised, the security requirement of signer ambiguity will not be fulfilled. In this paper, we propose an identity-based SDVS scheme which can resist the key-compromise attack. Compared with related works, the proposed scheme also has lower computational costs. Additionally, the security requirement of unforgeability against Existential Forgery on adaptive Chosen-Message Attacks (EF-CMA) is formally proved in the random oracle model.*

Keywords: *Identity-based, designated verifier, digital signature, key-compromise, bilinear pairing.*

Received Jun 27, 2012; accepted February 19, 2013; published online April 4, 2013

1. Introduction

In 1976, Diffie and Hellman [3] proposed the first public key cryptosystem based on the intractability of solving Discrete Logarithm Problem (DLP) [2]. In addition to public key encryption mechanisms, the digital signature scheme [4, 9, 15, 17] is another important technique of public key cryptosystems to satisfy essential security requirements such as integrity, authenticity [20] and non-repudiation [12]. Shamir [19] introduced the famous identity-based system in which each user's public key is explicitly his own identity and the corresponding private key is computed by a trusted authority. The derived private key is then sent to each user via a secure channel. Such a cryptosystem eliminates the possibility of public key substitution attacks and is extensively studied by many researchers since then. Consider some special application like the electronic voting [14, 18], in which the security requirement of non-repudiation is not desirable for guaranteeing the anonymity of vote.

To satisfy the application requirement, Chaum and Antwerpen [1] proposed an undeniable signature scheme in which the signature verification process can only be performed with the assistance of the signer. Namely, any verifier must obtain the signer's agreement in advance before validating the corresponding signature. In 1996, Jakobsson *et al*. [6] addressed the notion of designated verifier proofs and hence proposed the so-called Designated Verifier Signature (DVS) scheme. Unlike the undeniable signature scheme, a DVS scheme allows anyone to verify the signature without signer's agreement. However, only the designated verifier will be convinced of the signer's identity. Besides, the designated verifier cannot transfer the conviction to any third party, as he also has the ability to derive a valid DVS intended for himself with his own private key. Yet, in 2003 Wang [22] pointed out the security weakness of Jakobsson *et al*.'s scheme, i.e., a malicious signer can easily cheat the designated verifier.

In the same year, Saeednia *et al*. [16] further proposed a Strong Designated Verifier Signature (SDVS) scheme by incorporating the designated verifier's private key with the signature verification process, so as to prevent anyone else from validating the signature. In 2007, Lee and Chang [10] introduced SDVS schemes with message recovery to reduce the communication overheads. In 2009, Lee and Chang [11] also demonstrated that Saeednia *et al*.'s scheme could not fulfill the property of signer ambiguity in case that signer's private key is accidentally compromised. Generally speaking, an SDVS scheme should satisfy the following security requirements:

- *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.
- *Non-Transferability:* Since a designated verifier has the ability to generate another valid SDVS intended for himself with his own private key, he cannot transfer his conviction to any third party.
- *Signer Ambiguity:* It is difficult to determine the identity of signer for an intercepted SDVS which has not been received by the designated verifier.

Considering the identity-based systems, in 2004, Susilo

et al. [21] addressed the first identity-based SDVS scheme from bilinear pairings. Since then, several related works [5, 8, 23] have been proposed. In 2009, Kang *et al*. [7] proposed an identity-based SDVS scheme which has not only lower computational costs, but also shorter signature length. Nevertheless, their scheme is also vulnerable to the above key-compromise attack. Motivated by Kang *et al*.'s scheme, in this paper, we proposed an efficient and secure identity-based SDVS scheme which can resist the key-compromise attack. Moreover, the security requirement of unforgeability against Existential Forgery on adaptive Chosen-Message Attacks (EF-CMA) is formally proved in the random oracle model.

The rest of this paper is organized as follows: Section 2 states some preliminaries. We introduce the proposed identity-based SDVS scheme in section 3. The security proof and 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.

- *Bilinear Pairing:* Let $(G_1, +)$ and $(G_2, \times)$ denote two groups of the same prime order q and e: $G_1 \times G_1 \rightarrow G_2$ be a bilinear map which satisfies the following properties:
  1. *Bilinearity:*
     $e(P_1 + P_2, Q) = e(P_1, Q)\, e(P_2, Q)$;
     $e(P, Q_1 + Q_2) = e(P, Q_1)\, e(P, Q_2)$;
  2. *Non-degeneracy:*
     If P is a generator of G_1, then $e(P, P)$ is a generator of G_2.
  3. *Computability:*
     Given $P, Q \in G_1$, the value of $e(P, Q)$ can be efficiently computed by a polynomial-time algorithm.
- *Bilinear Diffie-Hellman Problem (BDHP):* The BDHP is, given an instance $(P, A, B, C) \in G_1^4$ where P is a generator, $A = aP$, $B = bP$ and $C = cP$ for some $a, b, c \in Z_q$, to compute $e(P, P)^{abc} \in G_2$.
- *Bilinear Diffie-Hellman (BDH) Assumption:* For every probabilistic polynomial-time algorithm A, every positive polynomial $Q(\cdot)$ and all sufficiently large k, the algorithm A can solve the BDHP with the advantage at most $1 / Q(k)$, i.e.,

$$Pr\,[A(P, aP, bP, cP) = e(P, P)^{abc};\ a, b, c \leftarrow Z_q,\ (P, aP, bP, cP) \leftarrow G_1^4] \leq 1 / Q(k).$$

The probability is taken over the uniformly and independently chosen instance and over the random choices of A.

- *Definition 1:* The (t, ε)-BDH assumption holds if there is no polynomial-time adversary that can solve the BDHP in time at most t and with the advantage ε.

3. The Proposed Scheme

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

3.1. Involved Parties

An identity-based SDVS scheme has three involved parties: A Private Key Generation center (PKG), a signer and a designated verifier. Each one is a Probabilistic Polynomial-Time Turing Machine (PPTM). The PKG is responsible for generating each user's private key and should be a trusted authority. The signer will generate an SDVS intended for the designated verifier. Consequently, the corresponding 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 will be convinced of the signer's identity.

3.2. Algorithms

The proposed identity-based SDVS scheme consists of the following algorithms:

- *Setup:* Taking as input 1^k where k is a security parameter, the PKG generates the system's public parameters *params*.
- *KeyExtract:* The KeyExtract algorithm takes as input the system parameters *params*, an identity *ID*, the master secret key of PKG. It generates the private key S_{ID}.
- *Signature-Generation (SG):* The SG algorithm takes as input the system parameters *params*, a message, the public key of designated verifier and the private key of signer. It generates the corresponding SDVSδ.
- *Signature-Verification (SV):* The SV algorithm takes as input the system parameters *params*, a message m, an SDVSδ, the private key of designated verifier and the public key of signer. It outputs True if 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, its SDVS δ and the private key of designated verifier. It outputs another valid SDVS δ^* for m.

3.3. Concrete Construction

Motivated by Kang *et al*.'s scheme [7], in this subsection, we give a concrete construction of our SDVS scheme. Details of each algorithm are described below:

- *Setup:* Taking as input 1^k, the PKG center chooses a master secret key $s \in_R Z_q$, computes the corresponding public key $P_{TA} = sP$ and then selects two groups $(G_1, +)$ and $(G_2, \times)$ of the same prime order q where $|q| = k$. Let P be a generator of order q over G_1, e: $G_1 \times G_1 \rightarrow G_2$ a bilinear pairing, H: $\{0, 1\}^k \rightarrow G_1$ and F:$\{0, 1\}^* \times G_2 \rightarrow Z_q$ collision resistant hash functions. The PKG announces public parameters $params = \{P_{TA}, G_1, G_2, q, P, e, H, F\}$.
- *KeyExtract:* Given an identity, say ID_a, the PKG computes the private key $S_a = sQ_a$, where $Q_a = H(ID_a)$ is the corresponding public key. The private key S_a is then delivered to the user U_a via a secure channel. U_a also announces a universal parameter $P_a = e(P, Q_a)$.
- *Signature-Generation (SG):* Let U_a be a signer and U_b associated with the identity ID_b the designated verifier. For signing a message $m \in \{0, 1\}^*$, U_a chooses $k \in_R Z_q$ to compute:

$$R = kP \tag{1}$$

$$C = P_b^k \tag{2}$$

$$D = k(P + P_{TA}) + F(m, C)\, S_a \tag{3}$$

$$W = e(D, Q_b) \tag{4}$$

And then sends m along with its SDVS $\delta = (W, R)$ to the designated verifier U_b.

- *Signature-Verification (SV):* Upon receiving (m, δ), U_b first computes:

$$C = e(R, Q_b) \tag{5}$$

And then verifies whether:

$$W = C \cdot e(R + F(m, C)\, Q_a, S_b) \tag{6}$$

If it holds, U_b is convinced that the SDVS $\delta = (W, R)$ for m is valid. Otherwise, an error symbol ¤ is returned to signal that δ is invalid. We show that the verification of equation 6 works correctly. From the right-hand side of equation 6, we have:

$$\begin{aligned}
&C \cdot e(R + F(m, C)\, Q_a, S_b) \\
&= e(P, kQ_b) \cdot e(R + F(m, C)\, Q_a, S_b) && \text{(by equations 1 and 5)} \\
&= e(P, kQ_b) \cdot e(kP + F(m, C)\, Q_a, S_b) && \text{(by equation 1)} \\
&= e(P, kQ_b) \cdot e(ksP + F(m, C)\, S_a, Q_b) \\
&= e(P, kQ_b) \cdot e(kP_{TA} + F(m, C)\, S_a, Q_b) \\
&= e(kP + kP_{TA} + F(m, C)\, S_a, Q_b) \\
&= e(D, Q_b) && \text{(by equation 3)} \\
&= W && \text{(by equation 4)}
\end{aligned}$$

Which leads to the left-hand side of equation 6.

- *Transcript-Simulation (TS):* To generate another SDVS δ^* intended for himself, U_b first chooses $R^* \in_R G_1$ and then computes:

$$C^* = e(R^*, Q_b) \tag{7}$$

$$W^* = C^* \cdot e(R^* + F(m, C^*)\, Q_a, S_b) \tag{8}$$

Here, $\delta^* = (W^*, R^*)$ is another valid SDVS for m.

4. Security Proof and Comparison

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

4.1. Security Requirement

The crucial security requirements of proposed SDVS scheme are non-transferability, signer ambiguity and unforgeability against EF-CMA. We define these notions as follows:

- *Definition 2:* An SDVS scheme is said to achieve the security requirement of unforgeability against EF-CMA if there is no probabilistic polynomial-time adversary A with non-negligible advantage in the following game played with a challenger B:
  - *Setup:* B first runs the Setup (1^k) algorithm and sends the system's public parameters *params* to the adversary A.
  - *Phase 1:* The adversary A can issue several queries adaptively, i.e., each query might be based on the result of previous queries:
    1. *KeyExtract Queries:* A makes a KeyExtract query for some identity ID. B returns the corresponding private key S_{ID}.
    2. *Signature-Generation (SG) Queries:* A makes an SG query for a message m with respect to a signer and a designated verifier. B returns the corresponding SDVS δ to A.
    3. *Signature-Verification (SV) Queries:* A gives B a message m and it's SDVS δ in relation to a signer and a designated verifier. If δ is a valid SDVS for m, B runs True. Otherwise, the error symbol $\perp$ is returned as a result.
- *Forgery:* Finally, A produces a new pair (m^*, δ^*) with the signer's identity ID_i^* and the designated verifier's identity ID_j^*. Note that A is not allowed to make a KeyExtract query for either ID_i^* or ID_j^*, and the SG query for m^* with the signer's identity ID_i^* and the designated verifier's identity ID_j^*. The adversary A wins if δ^* is a valid SDVS for m^*.
- *Definition 3:* An SDVS scheme satisfies the security requirement of signer ambiguity if there is no probabilistic polynomial-time adversary A having the ability to determine the identity of signer for an intercepted SDVS by performing the signature verification process before the SDVS has been received by the designated verifier.
- *Definition 4:* An SDVS scheme is said to achieve the security requirement of non-transferability if a designated verifier can simulate a computationally indistinguishable transcript intended for himself with his private key.

4.2. Security Proof

We prove that the proposed scheme achieves above defined security requirements below:

- *Theorem 1:* The proposed SDVS scheme is $(t, q_H, q_F, q_{KeyExtract}, q_{SG}, q_{SV}, \varepsilon)$ - secure against EF-CMA in the random oracle model if there is no probabilistic polynomial-time adversary that can (t', ε') - break the BDHP, where $\varepsilon' \geq 2(q_F^{-1})((\varepsilon - 2^{-k})(q_{SG}^2 - q_{SG})^{-1})^3$, and $t' \approx t + 2t_\lambda(2q_{SG} + q_{SV})$.

  Here, t_λ is the time for performing one bilinear pairing operation.
- *Proof:* Figure 1 depicts the proof structure of this theorem. Suppose that a probabilistic polynomial-time adversary A can $(t, q_H, q_F, q_{KeyExtract}, q_{SG}, q_{SV}, \varepsilon)$-break the proposed SDVS scheme with non-negligible advantage ε under adaptive chosen message attacks after running at most t steps and making at most q_H H, q_F F, $q_{KeyExtract}$ KeyExtract, q_{SG} SG and q_{SV} SV oracle queries. Then we can construct another algorithm B that can (t', ε')-break the BDHP by taking A as a subroutine. Let all involved parties and notations be defined the same as those in Section 3.3. The objective of B is to obtain $e(P, P)^{xyz}$ by taking $(P, q, e, X = xP, Y = yP, Z = zP)$ as inputs. In this proof, B simulates a challenger to A in the following game.

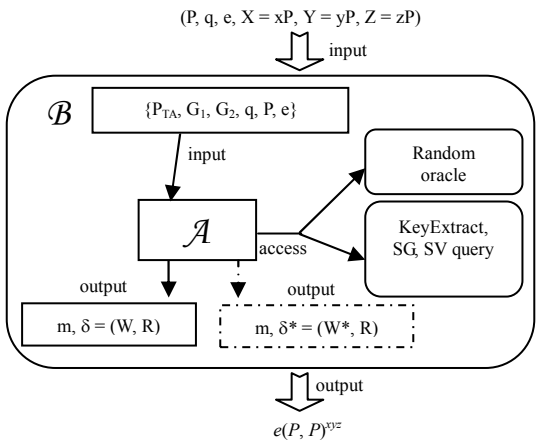

Figure 1. The proof structure of theorem 1.

- *Setup:* The challenger $\mathcal{B}$ runs the Setup (1^k) algorithm to obtain the system's public parameters $params = \{G_1, G_2, q, P, e\}$ and comes up with a random tape composed of a long sequence of random bits. Then $\mathcal{B}$ sets $P_{TA} = Z$. After that, $\mathcal{B}$ simulates two runs of SDVS scheme to the adversary $\mathcal{A}$ on input $\{P_{TA}, G_1, G_2, q, P, e\}$ and the random tape.
- *Phase 1:* $\mathcal{A}$ makes the following queries adaptively:
  1. *H Oracle:* When A queries an H oracle of $H(ID_i)$ with $i \neq (a, b)$, B first checks the H_list for a matched entry. Otherwise, B chooses $h_i \in_R Z_q$, adds the entry (ID_i, h_i, h_iP) to the H_list, and returns h_iP as a result. If $(i = a)$ or $(i = b)$, B directly returns X or Y to A, respectively.
  2. *F Oracle:* When A queries an F oracle of $F(m_i, C_i)$, B first checks the F_list for a matched entry. Otherwise, B chooses $f_i \in_R Z_q$ and adds the entry (m_i, C_i, f_i) to the F_list. Finally, B returns f_i as a result.
  3. *KeyExtract Queries:* When A makes a KeyExtract query for ID_i with $i \neq (a, b)$, B first searches the H_list for an entry containing ID_i, computes the corresponding private key $S_i = h_iZ$ and then returns S_i as a result. Otherwise, B terminates it.
  4. *SG Queries:* When A makes an SG query for some message m with the signer's identity ID_i and the designated verifier's identity ID_j where $j \neq (a, b)$, B first chooses $R \in_R G_1$, computes $C = e(R, Q_j)$ and $W = C \cdot e(R + F(m, C)Q_i, S_j)$, and then returns $\delta = (C, W, R)$ as the SDVS for m. Otherwise, B terminates it.
  5. *SV Queries:* When A makes an SV query for some pair (m, δ) with respect to the signer's identity ID_i and the designated verifier's identity ID_j where $j \neq (a, b)$, B first derives the designated verifier's private key $S_j = h_j Z$ and then follows the signature verification process in our proposed scheme to return the result. Otherwise, $\mathcal{B}$ terminates it.
- *Analysis of the Game:* Let Fv be the event that A attempts to forge an SDVS for some message m with the signer's identity ID_i and the designated verifier's identity ID_j and then finally outputs a valid forgery $\delta = (W, R)$. By assumption, we know that A has non-negligible probability ε to break the proposed SDVS scheme, i.e., $Pr[Fv] = \varepsilon$. The probability that A guesses the correct random value without asking $F(m, C)$ oracle is not greater than 2^{-k}. We denote such an event by $(\neg QF)$ and $Pr[\neg QF] \leq 2^{-k}$. Therefore, we can express the probability that A outputs a valid forgery after asking the corresponding F random oracle as:

$$Pr[Fv \wedge QF] \geq (\varepsilon - 2^{-k}) \quad (9)$$

Since $\mathcal{A}$ can ask at most q_{SG} SG queries, we will have $q_{SG}(q_{SG} - 1)$ pairs of the signer and the designated verifier. The probability that $(i, j) = \{(a, b) \text{ or } (b, a)\}$ is $2 / (q_{SG}(q_{SG} - 1))$. By combining equation 9, we can also derive that: $Pr[(Fv \wedge QF) \mid (i, j) = \{(a, b) \text{ or } (b, a)\}] \geq 2(\varepsilon - 2^{-k}) / (q_{SG}(q_{SG} - 1))$.

B again runs A on input $\{P_{TA}, G_1, G_2, q, P, e\}$ and the same random tape. Since A is supplied with the same sequence of random bits, we can expect that the i-th query he will ask is always the same as the one during the first simulation. For all the oracle queries before $F(m, C)$, B returns identical results as those in the first time. When A asks $F(m, C)$, B directly gives a

new $f^* \in_R Z_q$ instead of f. Meanwhile, A is then provided with another different random tape which is also composed of a long sequence of random bits. By the "Forking lemma" introduced by Pointcheval and Stern [13], if A eventually outputs another valid SDVS $\delta^* = (W^*, R)$ with $F(m, C) \neq F^*(m, C)$ and $(i, j) = \{(a, b)$ or $(b, a)\}$, B could obtain two equalities below:

$$W = C \cdot e(R + F(m, C)\, Q_a, S_b),$$
$$W^* = C \cdot e(R + F^*(m, C)\, Q_a, S_b)$$

Rewriting above two equalities, we have:

$$\begin{aligned}
& W \cdot e(R + F^*(m, C)\, Q_a, S_b) \\
& = W^* \cdot e(R + F(m, C)\, Q_a, S_b) \\
& \Rightarrow W/W^* = e(F(m, C)\, Q_a, S_b)\, e(F^*(m, C)\, Q_a, S_b)^{-1} \\
& \Rightarrow W/W^* = e(fQ_a, S_b)\, e(f^*Q_a, S_b)^{-1} \\
& \Rightarrow W/W^* = e((f - f^*)\, Q_a, S_b) \\
& \Rightarrow (W/W^*)^{(f-f^*)^{-1}} = e(Q_a, S_b) \\
& \Rightarrow (W/W^*)^{(f-f^*)^{-1}} = e(xP, z(yP)) \\
& \Rightarrow (W/W^*)^{(f-f^*)^{-1}} = e(P, P)^{xyz}
\end{aligned}$$

Consequently, B could solve the BDHP. To evaluate B's success probability, we use the "splitting lemma" [13] as follows:

Let Ω and Γ be the sets of possible sequences of random bits and random function values supplied to A before and after the $F(m, C)$ query is made by A, respectively. It follows that on inputting a random value $(\omega \parallel \gamma)$ for any $\omega \in \Omega$ and $\gamma \in \Gamma$, A outputs a valid forgery with the probability ε, i.e., $\Pr_{\omega \in \Omega,\, \gamma \in \Gamma}[\mathrm{Fv}] = \varepsilon$. According to the "splitting lemma", there exists a subset $N \in \Omega$ such that:

$$\Pr[\omega \in N] = |N| \cdot |\Omega|^{-1} \geq 2^{-1}\varepsilon$$
$$\forall \omega \in N, \Pr_{\gamma \in \Gamma}[\mathrm{Fv}] \geq 2^{-1}\varepsilon$$

From the above definition, we know that if $n \in N$ is the supplied sequence of random bits and random function values given to A before the $F(m, C)$ query is made, then for any sequence of random bits and random function values $\gamma' \in \Gamma$ after the query, A outputs a valid forgery with the probability of at least $(2^{-1}\varepsilon)^2 = 4^{-1}\varepsilon^2$, i.e., $Pr_{n \in N,\, \gamma' \in \Gamma}[Fv] \geq 4^{-1}\varepsilon^2$.

Since the probability that A outputs another SDVS $\delta^* = (W^*, R)$ with $F(m, C) \neq F^*(m, C)$ is q_F^{-1}, we can express the probability that B solve the BDHP after the second simulation as:

$$\varepsilon' \geq (2(\varepsilon - 2^{-k}) / (q_{SG}(q_{SG} - 1)))\, (4^{-1}(2(\varepsilon - 2^{-k}) / (q_{SG}(q_{SG} - 1)))^2)\, (q_F^{-1}) = 2(q_F^{-1})\, ((\varepsilon - 2^{-k})\, (q_{SG}^2 - q_{SG}^{-1})^3$$

Moreover, the computational steps required for B during the two simulations are $t' \approx t + 2t_\lambda(2q_{SG} + q_{SV})$ where t_λ is the time for performing one bilinear pairing operation.

- *Theorem 2:* The proposed SDVS scheme satisfies the security requirement of signer ambiguity even under the key-compromise attack.
- *Proof:* On the basis of the Signature-Verification (SV) algorithm in our proposed scheme, the signature verification equality, i.e., equation 10 can be further expressed as:

$$\begin{aligned}
W &= C \cdot e(R + F(m, C)\, Q_a, S_b) \\
&= C \cdot e(R + F(m, C)\, Q_a, Q_b)^s \\
&= C \cdot e(sR + F(m, C)\, S_a, Q_b) \\
&= C \cdot e(kP_{TA} + F(m, C)\, S_a, Q_b)
\end{aligned} \tag{10}$$

It is obvious that even if the signer's private key S_a is compromised, any malicious adversary still needs the knowledge of secret integer k to perform equation 7. Hence, the signer ambiguity is fulfilled in the proposed scheme even under the key-compromise attack.

- *Theorem 3:* The proposed SDVS scheme satisfies the security requirement of non-transferability.
- *Proof:* By the Transcript-Simulation (TS) algorithm of our scheme, any designated verifier can easily derive another valid SDVS δ^* intended for himself with his private key. The generated SDVS δ^* is computationally indistinguishable from the received δ. To be precise, the probability that the computed $\delta^* = (W^*, R^*)$ and the received $\delta = (W, R)$ are identical is at most $|G_1|^{-1}$, i.e., $\Pr[\delta^* = \delta] \leq |G_1|^{-1}$.

4.3. Comparisons

We compare our proposed scheme with several related ones including Kumar *et al.*'s (KSS for short) [8], Susilo *et al.*'s (SZM for short) [21], the Zhang-Mao (ZM for short) [23] and Kang *et al.*'s (KBD for short) [7] schemes. Note that the computational costs are evaluated in number of the most time-consuming two operations, i.e., the bilinear pairing and the exponentiation computations. For facilitating the following comparisons, we define some used notations:

$|x|$: The bit-length of an integer x
B: A bilinear pairing computation
E: An exponentiation computation in G_2

Detailed comparisons are demonstrated in Table 1. In this comparison, we assume that $|G_1| = |G_2|$ and the computational costs include signature generation and verification. From this table, it can be seen that only Kumar *et al.*'s scheme and ours can resist the key-compromise attack. However, their scheme incurs higher computational costs and longer signature length. As a whole, we conclude that the proposed identity-based SDVS scheme not only is secure against the key-compromise attack, but also provides better efficiency.

Table 1. Comparisons of the proposed and related schemes.

Item / Scheme	Resist Key-Compromise Attack	Computational Costs[1]	Signature Length[2]
KSS	√	5B	$4\|G_1\|$
SZM	×	3B + 3E	$\|G_1\| + 2\|q\|$
ZM	×	3B	$3\|G_1\|$
KBD	×	3B + 2E	$2\|G_1\|$
Ours	√	3B + E	$2\|G_1\|$

5. Conclusions

SDVS scheme is a special type of digital signature schemes, which has crucial benefits to the application such as the electronic voting. In this paper, we focused on the ID-based cryptosystem and proposed an efficient identity-based SDVS scheme which is secure against the key-compromise attack. The underlining security assumption of our proposed scheme is the well-known BDHP which is believed to be unsolvable in polynomial time. We showed that the proposed scheme fulfills all the essential security requirements. Besides, the unforgeability against EF-CMA is formally proved in the random oracle model. Compared with previous related works, our proposed SDVS scheme not only is more secure, but also earns more computational efficiency.

Acknowledgement

We would like to thank anonymous referees for their valuable suggestions. This work was supported in part by the National Science Council of Republic of China under the contract number NSC 101-2218-E-019-005.

References

[1] Chaum D. and Antwerpen H., "Undeniable Signature," *in Proceedings of Advances in Cryptology, Lecture Notes in Computer Science*, Berlin, Germany, vol. 435, pp. 212-216, 1990.

[2] Delfs H. and Knebl H., *Introduction to Cryptography: Principles and Applications*, Springer, Berlin, Germany, 2002.

[3] Diffie W. and Hellman M., "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644-654, 1976.

[4] ElGamal T., "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469-472, 1985.

[5] Huang X., Susilo W., Mu Y., and Zhang F., "Short Designated Verifier Signature Scheme and its Identity-Based Variant," *the International Journal of Network Security*, vol. 6, no. 1, pp. 82-93, 2008.

[6] Jakobsson M., Sako K., and Impagliazzo R., "Designated Verifier Proofs and their Applications," *in Proceedings of International Conference on the Theory and Application of Cryptographic Techniques Saragossa*, Spain, pp. 143-154, 1996.

[7] Kang B., Boyd C., and Dawson E., "A Novel Identity-Based Strong Designated Verifier Signature Scheme," *the Journal of Systems and Software*, vol. 82, no. 2, pp. 270-273, 2009.

[8] Kumar K., Shailaja G., and Saxena A., "Identity Based Strong Designated Verifier Signature Scheme," available at: http://eprint.iacr.org/2006/134.pdf, last visited 2006.

[9] Lakshmanan T. and Muthusamy M., "A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes," *the International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 262-267, 2012.

[10] Lee J. and Chang J., "Strong Designated Verifier Signature Scheme with Message Recovery," *in Proceedings of the 9th International Conference on Advanced Communication Technology*, Gangwon-Do, South Korea, vol. 1, pp. 801-803, 2007.

[11] Lee J. and Chang J., "Comment on Saeednia *et al.*'s Strong Designated Verifier Signature Scheme," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 258-260, 2009.

[12] Meng B., Wang S., and Xiong Q., "A Fair Non-Repudiation Protocol," *in Proceedings of the 7th International Conference on Computer Supported Cooperative Work in Design*, Brazil, pp. 68-73, 2002.

[13] Pointcheval D. and Stern J., "Security Arguments for Digital Signatures and Blind Signatures," *Journal of Cryptology*, vol. 13, no. 3, pp. 361-369, 2000.

[14] Ray I. and Narasimhamurthi N., "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*, San Juan, USA, pp. 188-190, 2001.

[15] Rivest R., Shamir A., and Adleman L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[16] Saeednia S., Kremer S., and Markowitch O., "An Efficient Strong Designated Verifier Signature Scheme," *in Proceedings of the 6th International Conference on Information Security and Cryptology*, Seoul, Korea, pp. 40-54, 2003.

[17] Schnorr C., "Efficient Signature Generation by Smart Cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161-174, 1991.

[18] Schoenmakers B., "A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting," *in Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, London, United Kingdom, pp. 148-164, 1999.

[19] Shamir A., "Identity-Based Cryptosystems and Signature Schemes," *in Proceedings of Advances in Cryptology, Lecture Notes in Computer Science*, Berlin, Germany, vol. 196, pp. 47-53, 1984.

[20] Stallings W., *Cryptography and Network Security: Principles and Practices*, Pearson Hall, New York, USA, 2005.

[21] Susilo W., Zhang F., and Mu Y., "Identity-Based Strong Designated Verifier Signature Schemes," *in Proceedings of the 9th Australasian Conference on Information Security and Privacy, Lecture Notes in Computer Science*, Sydney, Australia, pp. 313-324, 2004.

[22] Wang G., "An Attack on Not-Interactive Designated Verifier Proofs for Undeniable Signatures," *Cryptology ePrint Archive*, available at: http://eprint.iacr.org/2003/243, last visited 2003.

[23] Zhang J. and Mao J., "A Novel ID-Based Designated Verifier Signature Scheme," *Information Sciences*, vol. 178, no. 3, pp. 766-773, 2008.

Han-Yu Lin received his BA degree in economics from the Fu-Jen University, Taiwan in 2001, his MS degree in information management from the Huafan University, Taiwan in 2003, and his PhD degree in computer science and engineering from the National Chiao Tung University, Taiwan in 2010. Since August 2012, he has been an Assistant Professor in the Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan.His research interests include cryptology, network security and e-commerce security.