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.

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

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

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

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 SDVS \(\delta\).

- **Signature-Verification (SV):** The SV algorithm takes as input the system parameters \(params\), a message \(m\), an SDVS \(\delta\), the private key of designated verifier and the public key of signer. It outputs True if \(\delta\) 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 \(\delta\) and the private key of designated verifier. It outputs another valid SDVS \(\delta^*\) 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:
Toward Secure Strong Designated Verifier Signature Scheme from Identity-Based System

- Setup: Taking as input $1^k$, the PKG center chooses a master secret key $s \in \mathbb{Z}_p^*$, computes the corresponding public key $P_{sk} = sP$ 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 $F: \{0, 1\}^* \times G_2 \rightarrow \mathbb{Z}_q$ collision resistant hash functions. The PKG announces public parameters $\text{params} = \{P_{sk}, G_1, G_2, q, P, e, H, F\}$.

- 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 $k \in \mathbb{Z}_q^*$ to compute:
  \[
  R = kP \quad (1)
  \]
  \[
  C = P_b^k \quad (2)
  \]
  \[
  D = k(P + P_{sk}) + F(m, C)S_a \quad (3)
  \]
  \[
  W = e(D, Q_b) \quad (4)
  \]
  And then sends $m$ along with its SDVS $\delta = (W, R)$ to the designated verifier $U_b$.

- Signature-Verification (SV): Upon receiving $(m, \delta)$, $U_b$ first computes:
  \[
  C = e(R, Q_b) \quad (5)
  \]
  And then verifies whether:
  \[
  W = C \cdot e(R + F(m, C)Q_a, S_a) \quad (6)
  \]
  If it holds, $U_b$ is convinced that the SDVS $\delta = (W, R)$ for $m$ is valid. Otherwise, an error symbol $\perp$ is returned to signal that $\delta$ is invalid. We show that the verification of equation 6 works correctly. From the right-hand side of equation 6, we have:
  \[
  C \cdot e(R + F(m, C)Q_a, S_a) = e(P, kQ_a) \cdot e(R + F(m, C)Q_a, S_a) \quad (by \ equations \ 1 \ and \ 5)
  \]
  \[
  = e(P, kQ_a) \cdot e(kP + F(m, C)Q_a, S_a) \quad (by \ equation \ 1)
  \]
  \[
  = e(P, kQ_a) \cdot e(kP + F(m, C)S_a, Q_b) \quad (by \ equation \ 3)
  \]
  \[
  = e(D, Q_b) \quad (by \ equation \ 4)
  \]
  Which leads to the left-hand side of equation 6.

- Transcript-Simulation (TS): To generate another SDVS $\delta^*$ intended for himself, $U_b$ first chooses $R^* \in \mathbb{Z}_q^*$ and then computes:
  \[
  C^* = e(R^*, Q_b) \quad (7)
  \]
  \[
  W^* = C^* \cdot e(R^* + F(m, C^*)Q_a, S_a) \quad (8)
  \]
  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 $B$:
  \[
  \begin{align*}
  \text{Setup: } & B \text{ first runs the Setup (1)}^k \text{ algorithm and sends the system’s public parameters } \text{params} \text{ to the adversary } A. \\
  \text{Phase 1: } & A \text{ can issue several queries adaptively, i.e., each query might be based on the result of previous queries:} \\
  & 1. \text{KeyExtract Queries: } A \text{ makes a KeyExtract query for some identity } ID_{a}. B \text{ returns the corresponding private key } S_{ID_{a}}. \\
  & 2. \text{Signature-Generation (SG) Queries: } A \text{ makes an SG query for a message } m \text{ with respect to a signer and a designated verifier. } B \text{ returns the corresponding SDVS } \delta \text{ to } A. \\
  & 3. \text{Signature-Verification (SV) Queries: } A \text{ gives } B \text{ a message } m \text{ and it’s SDVS } \delta \text{ in relation to a signer and a designated verifier. If } \delta \text{ is a valid SDVS for } m, B \text{ runs True. Otherwise, the error symbol } \perp \text{ is returned as a result.}
  \end{align*}
  \]

- Forgery: Finally, $A$ produces a new pair $(m^*, \delta^*)$ with the signer’s identity $ID_{a^*}$ and the designated verifier’s identity $ID_{b^*}$. Note that $A$ is not allowed to make a KeyExtract query for either $ID_{a^*}$ or $ID_{b^*}$, and the SG query for $m^*$ with the signer’s identity $ID_{a^*}$ and the designated verifier’s identity $ID_{b^*}$. The adversary $A$ wins if $\delta^*$ 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 \((t', \varepsilon')\)-break the BDHP, where \(\varepsilon' \geq 2q_F' \cdot \left(\varepsilon - 2^{-2}q_{SG} - q_{SG}^{-1}\right)\), and \(t' = t + 2t_A(2q_{SG} + q_{SG})\).

Here, \(t_j\) 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 \(\varepsilon\) under adaptive chosen message attacks after running at most \(t\) steps and making at most \(q_F H, q_F, q_{KeyExtract, KeyExtract, SG}, q_{SV} SG, q_{SV} SV\) oracle queries. Then we can construct another algorithm \(B\) that can \((t', \varepsilon')\)-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 \(e(P, P)^{SV}\) 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.

\[
\begin{align*}
(P, q, e, X = xP, Y = yP, Z = zP) & \quad \text{input} \\
B & \quad \text{Random oracle} \\
A & \quad \text{KeyExtract, SG, SV query} \\
& \downarrow \text{access} \\
& m, \delta = (W, R) \\
& \text{output} \\
& \text{output} \\
& e(P, P)^{SV} \\
& \downarrow \text{input}
\end{align*}
\]

Figure 1. The proof structure of theorem 1.

- **Setup**: The challenger \(B\) 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 \(B\) sets \(P_{TA} = Z\). After that, \(B\) simulates two runs of SDVS scheme to the adversary \(A\) on input \(\{P_{TA}, G_1, G_2, q, P, e\}\) and the random tape.

- **Phase 1**: \(A\) makes the following queries adaptively:
  1. **H Oracle**: When \(A\) queries an \(H\) oracle of \(H(\text{ID})\) with \(i \neq (a, b)\), \(B\) first checks the \(H\) list for a matched entry. Otherwise, \(B\) chooses \(h_i \in \mathcal{R} Z_q\), adds the entry \((\text{ID}_i, h_i, h_P)\) to the \(H\) list, and returns \(h_P\) 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 \(F(m, C)\), \(B\) first checks the \(F\) list for a matched entry. Otherwise, \(B\) chooses \(f_i \in Z_q\) and adds the entry \((m_i, C, f_i)\) to the \(F\) list. Finally, \(B\) returns \(f_i\) as a result.
  3. **KeyExtract Queries**: When \(A\) makes a KeyExtract query for \(ID\), with \(i \neq (a, b)\), \(B\) first searches the \(H\) list for an entry containing \(ID_i\), computes the corresponding private key \(S_i = h_i Z\) 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 \(ID\) and the designated verifier’s identity \(ID_j\) where \(j \neq (a, b)\), \(B\) first chooses \(R_i \in \mathcal{R} G_1\), computes \(C = e(R, Q_j)\) and \(W = C \cdot e(R + F(m, C), Q_j, S_j)\), 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 \((m, \delta)\) with respect to the signer’s identity \(ID\) 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, \(B\) terminates it.

- **Analysis of the Game**: Let \(Fv\) be the event that \(A\) attempts to forge a SDVS for some message \(m\) with the signer’s identity \(ID\) 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 \(\varepsilon\) 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 \((-QF)\) and \(Pr[¬QF] \leq 2^{-k}\). Therefore, we can express the probability that \(A\) outputs a valid forgery after asking the corresponding \(F\) random oracle as:

\[
Pr[Fv \land QF] \geq (\varepsilon - 2^{-k}) 
\]

Since \(A\) 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 \((i, j) = \{(a, b)\} or (b, a)\) is \(2 / (q_{SG} (q_{SG} - 1))\). By combining equation 9, we can also derive that: \(Pr[\{Fv \land QF\} | (i, j) = \{(a, b)\} or (b, a)\}] \geq 2(\varepsilon - 2^{-k}) / (q_{SG} (q_{SG} - 1))\).

\(B\) 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 \( f^* \in \mathbb{R} \) 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:

\[
W = C \cdot e (R + F(m, C) Q_y, S_y), \quad W^* = C \cdot e (R + F^*(m, C) Q_y, S_y)
\]

Rewriting above two equalities, we have:

\[
W \cdot e (R + F^*(m, C) Q_y, S_y) = W^* \cdot e (R + F(m, C) Q_y, S_y) \\
\Rightarrow \frac{W}{W^*} = e (F(m, C) Q_y, S_y) e (F^*(m, C) Q_y, S_y)^{-1} \\
\Rightarrow \frac{W}{W^*} = e (f Q_y, S_y) e (f^* Q_y, S_y)^{-1} \\
\Rightarrow \frac{W}{W^*} = e ((f - f^*) Q_y, S_y) \\
\Rightarrow \frac{W}{W^*} = e (y P, z(y P)) \\
\Rightarrow \frac{W}{W^*} = e (P, y)\]

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

Let \( \Omega \) and \( \Gamma \) 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 \( (\omega || \gamma) \) for any \( \omega \in \Omega \) and \( \gamma \in \Gamma \), \( A \) outputs a valid forgery with the probability \( \varepsilon \), i.e., \( \Pr_{\omega \in \Omega, \gamma \in \Gamma} [Fv = \varepsilon] \). According to the “splitting lemma”, there exists a subset \( N \in \mathbb{R} \) such that:

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

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 \( (2^{-1} \varepsilon)^2 = 4^{-1} \varepsilon \), i.e., \( \Pr_{\varepsilon \in \Omega, \gamma' \in \Gamma} [Fv] \geq 4^{-1} \varepsilon \).

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

\[
\varepsilon' \geq (2 (2 \varepsilon^2 / (q_{SG}(q_{SG}-1))) (4^{-1} (2 (2 \varepsilon^2) / (q_{SG}(q_{SG}-1))))^2 (q_{SG}^{-1})^2 = 2q_{SG}^{-1} ((\varepsilon - 2 \varepsilon^2) (q_{SG}^{-2} - q_{SG}^{-3}))
\]

Moreover, the computational steps required for \( B \) during the two simulations are \( t' \approx t + 2t_s (2q_{SG} + q_{SD}) \) where \( t_s \) 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:

\[
W = C \cdot e (R + F(m, C) Q_y, S_y) \\
= C \cdot e (R + F(m, C) Q_y, Q_0)^{t'} \\
= C \cdot e (t R + F(m, C) S_y, Q_0) \\
= C \cdot e (t R P + F(m, C) S_y, Q_0)
\]

It is obvious that even if the signer’s private key \( S_y \) 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 \( \delta^* \) intended for himself with his private key. The generated SDVS \( \delta^* \) is computationally indistinguishable from the received \( \delta \). To be precise, the probability that the computed \( \delta^* = (W^*, R^* \rangle \) 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:

\[
|x|: \text{The bit-length of an integer } x \\
B: \text{A bilinear pairing computation} \\
E: \text{An exponentiation computation in } G_2
\]

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.

<table>
<thead>
<tr>
<th>Item Scheme</th>
<th>Resist Key-Compromise Attack</th>
<th>Computational Costs1</th>
<th>Signature Length2</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSS</td>
<td>✓</td>
<td>5B</td>
<td>4G1</td>
</tr>
<tr>
<td>SZM</td>
<td>×</td>
<td>3B + 3E</td>
<td>3G2</td>
</tr>
<tr>
<td>ZM</td>
<td>×</td>
<td>3B</td>
<td>3G2</td>
</tr>
<tr>
<td>KBD</td>
<td>×</td>
<td>3B + 2E</td>
<td>2G2</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>3B + E</td>
<td>2G2</td>
</tr>
</tbody>
</table>
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 underlying 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


**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.