Communications of the CCISA
Vol. 22 No. 3 Jul. 2016

Selectively Secure Lightweight Strong Designated Verifier Signature Scheme from Identity-Based System

Han-Yu Lin[1*] and Yao-Min Hung[2]
Department of Computer Science and Engineering,
National Taiwan Ocean University,
2, Beining Road, Keelung, Taiwan
[1]lin.hanyu@msa.hinet.net, [2]n7773246@gmail.com

ABSTRACT

In some special electronic applications like personal medical records, authenticity and privacy are considered the basic security requirements. A strong designated verifier signature (SDVS) scheme is applicable to the above scenarios. Such a scheme allows a signer to create a so-called designated signature which can only be verified by an intended verifier. Besides, the designated verifier has no way to transfer his conviction to any third party. It is thus can be seen that SDVS schemes play an important role in privacy-preserving applications. In this paper, we propose a lightweight SDVS scheme from identity-based systems. More specifically, our scheme owns lower computational costs as compared with previous mechanisms. Furthermore, the selective security against universal forgery attack is also realized in the random oracle model.

Keywords: strong designated verifier, privacy-preserving, identity-based, lightweight, digital signature.

1. Introduction

To withstand the public key substitution attacks and solve the certificate management problem, in 1984, Shamir [11] introduced the famous identity-based cryptosystem. In this system, each user's public key is his/her public identifier that can be explicitly authenticated without extra verification processes. A system authority (SA) is responsible for deriving all users' private keys with a trapdoor one-way function and then sends the private key to each user via a secure channel. Therefore, it is difficult for any adversary to compute the corresponding private key from its public one without knowing the trapdoor.

Considering the requirement of privacy-preserving applications, in 1990, Chaum and Antwerpen [1] presented a special type of signatures called undeniable signature. Such a

* Corresponding Author: Han-Yu Lin

signature scheme provides the signer with the right to decide who can verify his generated signatures. That is, a verifier must obtain the original signer's agreement and cooperatively his signatures.

In 1996, Jakobsson *et al*. [3] further addressed the concept of designated verification and proposed a Designated Verifier Signature (DVS) scheme with the property of non-transferability. In a DVS scheme, an intended verifier is able to simulate another transcript with his/her private key. Consequently, it is difficult for anyone to identify the real signer from two candidates. Only the intended verifier of the signature will be convinced of the signer's identity. However, in 2003, Wang [13] pointed out some security flaws of their scheme.

In the same year, Saeednia *et al*. [10] introduced the notion of strong DVS (SDVS) schemes in which the signature verification process can only be performed with the assistance of the designated verifier's private key. An SDVS scheme still has the property of non-transferability, i.e., a designated verifier cannot convince any third party of the signer's identity, since he can also create a computationally indistinguishable transcript.

In 2004, Susilo *et al*. [12] combined the SDVS scheme with identity-based cryptosystems to propose the first identity-based SDVS scheme using the Bilinear Diffie-Hellman Problem (BDHP). In 2007, Lee and Chang [6] introduced an SDVS variant in which the designated verifier can recover the original message from its signature. Consequently, the signed message is unnecessary to be transmitted along with the signature. To reduce the signature length, in 2009, Kang *et al*. [4] also addressed an SDVS variant. In 2011, Lin *et al*. [8] proposed a new SDVS scheme using the Discrete Logarithm Problem (DLP). Their scheme outperforms related works in terms of computational costs and signature length. Up to present, several SDVS variants [2, 5, 14] have been introduced.

In this paper, we consider the bilinear pairing cryptosystems from elliptic curves and will present an identity-based lightweight SDVS scheme. In our scheme, we attempt to reduce the operation of bilinear pairing computation, so as to gain more savings of computational costs. Additionally, we will also demonstrate that the proposed mechanism fulfills the selective security against universal forgery attacks in the random oracle model.

2. Preliminaries

In this section, we review the properties of bilinear pairing and its security assumption.

Bilinear Pairing

Let $(\boldsymbol{G}_1, +)$ and $(\boldsymbol{G}_2, \times)$ separately denote two groups of prime order q and e: $\boldsymbol{G}_1 \times \boldsymbol{G}_1 \rightarrow$

$\boldsymbol{G}_2$ be a bilinear map satisfying the following properties:

(i) ***Bilinearity***:

$e(P_1 + P_2, Q) = e(P_1, Q)e(P_2, Q)$;

$e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2)$;

(ii) ***Non-degeneracy***:

If P is a generator of $\boldsymbol{G}_1$, then $e(P, P)$ is a generator of $\boldsymbol{G}_2$.

(iii)***Computability***:

Given $P, Q \in \boldsymbol{G}_1$, the value of $e(P, Q)$ can be efficiently computed by a polynomial-time algorithm.

Bilinear Diffie-Hellman Problem; BDHP

Given $P, aP, bP, cP \in \boldsymbol{G}_1$ for some $a, b, c \in Z_q$, the bilinear Diffie-Hellman problem is to compute $e(P, P)^{abc} \in \boldsymbol{G}_2$.

Bilinear Diffie-Hellman (BDH) Assumption

For every probabilistic polynomial-time algorithm $\mathcal{A}$, every positive polynomial $F(\cdot)$ and all sufficiently large k, the algorithm $\mathcal{A}$ can solve the BDHP with an advantage of at most $1/F(k)$, i.e.,

$\Pr[\mathcal{A}(P, aP, bP, cP) = e(P, P)^{abc}; a, b, c \leftarrow Z_q, (P, aP, bP, cP) \leftarrow \boldsymbol{G}_1{}^4] \leq 1/F(k)$.

The probability is taken over the uniformly and independently chosen instance and over the random choices of $\mathcal{A}$.

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 an advantage* ε.

3. The Proposed Scheme

We first define involved parties and algorithms of our scheme and then give a concrete construction.

3.1 Involved Parties

In the proposed SDVS scheme, there are three parties including a system authority (SA), a signer and a designated verifier. All of the three parties is a probabilistic polynomial-time Turing machine (PPTM). The SA is responsible for generating system parameters and distributes the private key to every user. The signer can incorporate a designated verifier's public key with his private key to create an SDVS intended for the designated verifier. After

receiving the SDVS, the designated verifier can authenticate it with his private key. Meanwhile, it is unable for the designated verifier to convince any third party of his proof.

3.2 Algorithms

The proposed scheme is composed of four algorithms stated as follows:

– **Setup:** Taking as input a security parameter k, the algorithm outputs corresponding public parameters *params*.

– **SDVS-Generation (SDVS-G):** The SDVS-G algorithm takes as input the system parameters *params*, a message m, the public key of designated verifier and the private key of signer. It generates an SDVS δ.

– **SDVS-Verification (SDVS-V):** The SDVS-V algorithm takes as input the system parameters *params*, a message m, an SDVS δ, the private key of designated verifier and the public key of signer. It outputs **True** if δ is a valid SDVS for m. Otherwise, an error 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 Construction

We present a concrete construction of the proposed scheme as follows:

– **Setup:** Taking as input a security parameter k, the SA chooses two groups $(\boldsymbol{G}_1, +)$ and $(\boldsymbol{G}_2, \times)$ of prime order q for $|q| = k$. Let P be a generator of order q over $\boldsymbol{G}_1$, e: $\boldsymbol{G}_1 \times \boldsymbol{G}_1 \rightarrow \boldsymbol{G}_2$ a bilinear pairing and H_1: $\{0, 1\}^* \rightarrow \boldsymbol{G}_1$ and H_2: $\{0, 1\}^* \times \boldsymbol{G}_1 \rightarrow Z_q$ two secure hash functions. The system parameters *params* is $\{\boldsymbol{G}_1, \boldsymbol{G}_2, q, P, e, H_1, H_2\}$. The SA selects s as its master private key and the public key is computed as $P_{SA} = sP$. Each user U_i's key pair is $(S_i = sQ_i, Q_i = H_1(ID_i))$.

– **SDVS-Generation (SDVS-G):** For signing a message $m \in_R \{0, 1\}^*$, a signer U_A first chooses $r \in_R Z_q$ to compute

$$K = rP_{SA}, \tag{1}$$

$$Z = e(Q_B - rP, H_2(m, K)S_A), \text{ where } Q_B \text{ is the designated verifier.} \tag{2}$$

The SDVS for the message m is $\delta = (K, Z)$.

– **SDVS-Verification (SDVS-V):** In order to verify the SDVS $\delta = (K, Z)$, U_B can check whether

$$Z = e(S_B - K, H_2(m, K)Q_A). \quad (3)$$

If the quality holds, the SDVS for m is valid. We show that the verification of Eq. (3) works correctly. From the left-hand side of Eq. (3), we have

$$\begin{aligned} & Z \\ &= e(Q_B - rP, H_2(m, K)S_A) && \text{(by Eq. (2))} \\ &= e(Q_B, H_2(m, K)S_A)/e(rP, H_2(m, K)S_A) \\ &= e(Q_B, H_2(m, K)S_A)/e(K, H_2(m, K)Q_A) && \text{(by Eq. (1))} \\ &= e(S_B, H_2(m, K)Q_A)/e(K, H_2(m, K)Q_A) \\ &= e(S_B - K, H_2(m, K)Q_A) \end{aligned}$$

which leads to the right-hand side of Eq. (3).

– **Transcript-Simulation (TS):** To produce a transcript intended for himself, U_B chooses a new $K' \in_R G_1$ to compute

$$Z' = e(S_B - K', H_2(m, K')Q_A). \quad (4)$$

Then, $\delta' = (K', Z')$ is another valid SDVS for m.

4. Security Proof and Evaluation

In this section, we first define the security requirements and then prove the security of our scheme. In addition, some comparisons with previous works are also made.

Definition 2. *An SDVS scheme is said to achieve the security requirement of unforgeability against existential forgery if there is no probabilistic polynomial-time adversary $\mathcal{A}$ with non-negligible advantage in the following game played with a challenger $\mathcal{B}$:*

Setup: $\mathcal{B}$ 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 adaptively ask hash random oracles, i.e., each query might be based on the result of previous queries:

Forgery: $\mathcal{A}$ produces a pair (m^*, δ^*) with the signer's identity ID_A^* and the designated verifier's identity ID_B^*. The adversary $\mathcal{A}$ 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 $\mathcal{A}$ that can determine the identity of signer for a given SDVS with respect to two candidate signers.*

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.

Theorem 1. *The proposed ID-based SDVS scheme is* $(t, q_{H_1}, q_{H_2}, \varepsilon)$*-selectively secure against universal forgery attacks in the random oracle model if there is no probabilistic polynomial-time adversary* $\mathcal{A}$ *that can break the BDHP with non-negligible advantage.*

Proof: Assume that there is a probabilistic polynomial-time adversary $\mathcal{A}$ who can forge a valid SDVS of our proposed scheme with the non-negligible advantage after making at most q_{H_i} H_i random oracles (for i = 1 and 2). Then by utilizing $\mathcal{A}$ as a subroutine, it is feasible for us to generate another algorithm $\mathcal{B}$ solving the BDHP. Let all involved parties and notations be defined the same as those in Section 3. The goal of $\mathcal{B}$ is to outpute $e(P, P)^{abc}$ by taking the BDHP instance (P, aP, bP, cP) as inputs. In this proof, we employ the technique of Forking Lemma [9] to prove this theorem. Let $\mathcal{B}$ simulate a challenger to $\mathcal{A}$ in the following game.

Setup: The challenger $\mathcal{B}$ first performs the Setup algorithm to obtain $params = \{\boldsymbol{G}_1, \boldsymbol{G}_2, q, P, e\}$ and prepares a random tape Σ consisting of a long sequence of random bits. Then, $\mathcal{B}$ sets $P_{SA} = aP$ and simulates two runs of the proposed scheme to $\mathcal{A}$ with the input values ($params$, Σ, P_{SA}).

Phase 1: $\mathcal{A}$ can query the following random oracles adaptively:

– H_1 *oracle:* When $\mathcal{A}$ queries $H_1(ID_i)$ oracle, $\mathcal{B}$ searches the maintained H_1_list for a matched entry. Otherwise, $\mathcal{B}$ chooses $v_1 \in_R Z_q$, stores the record of (ID_i, v_1, v_1P) into H_1_list and returns v_1P as a result. Note that if $ID_i \in \{U_A, U_B\}$, $\mathcal{B}$ directly returns $\{bP, cP\}$, i.e., implicitly define $Q_A = bP$ and $Q_B = cP$.

– H_2 *oracle:* When $\mathcal{A}$ makes the $H_2(m, K)$ query, $\mathcal{B}$ first checks the maintained H_2_list for a possible record. Otherwise, $\mathcal{B}$ chooses $v_2 \in_R Z_q$, stores the entry (m, K, v_2) into H_2_list and returns v_2 as a result.

Forgery: Finally, $\mathcal{A}$ produces a forgery $\delta = (K, Z)$ for his arbitrarily chosen message m.

Analysis of the game: $\mathcal{B}$ again simulates the second run with $\mathcal{A}$ on the same input. Since we supply the adversary $\mathcal{A}$ with the identical random tape, he will query the same oracles as those during the first run. When $\mathcal{A}$ asks the critical $H_2(m, K)$ oracle this time, $\mathcal{B}$ returns a new answer v_2* instead of original v_2. By the "Forking lemma", when $\mathcal{A}$ at last generates a different forgery δ* = (K, Z*) with $H_2(m, K) = v_2$*, $\mathcal{B}$ will have two equalities below:

$$Z = e(S_B - K, v_2 Q_A),$$
$$Z^* = e(S_B - K, v_2^* Q_A).$$

Further rewriting these equalities, we will learn that

$$Z/Z^* = e(S_B - K, (v_2 - v_2^*)Q_A)$$

$= e(acP - K, (v_2 - v_2^*)bP)$

$= e(acP, (v_2 - v_2^*)bP)/e(K, (v_2 - v_2^*)bP)$

$= e(abcP, (v_2 - v_2^*)P)/e(K, (v_2 - v_2^*)bP)$

$\Rightarrow \quad (Z/Z^*)e(K, (v_2 - v_2^*)bP) = e(P, P)^{abc(v_2 - v_2^*)}$.

Hence, $\mathcal{B}$ could solve the BDHP instance by computing

$$e(P,P)^{abc} = [(Z / Z^*)e(K,(v_2 - v_2^*)bP)]^{(v_2 - v_2^*)^{-1}}.$$

Q.E.D.

Theorem 2. *The proposed ID-based SDVS scheme satisfies the security requirement of signer ambiguity even under the key-compromise attack.*

Proof: This proof demonstrates that even if an adversary has the knowledge of signer's private key, he still cannot distinguish the real signer from a designated verifier for a given SDVS. In our scheme, Eq. (3), can be derived as

$Z = e(S_B - K, H_2(m, K)Q_A)$

$= e(Q_B, H_2(m, K)S_A)/e(rP, H_2(m, K)S_A)$.

It is obvious that any adversary has to know the random number r before he can compute this equality. Hence, our scheme satisfies the property of signer ambiguity even under the key-compromise attack.

Q.E.D.

Theorem 3. *The proposed ID-based SDVS scheme satisfies the security requirement of non-transferability.*

Proof: In the Transcript-Simulation (TS) algorithm, a designated verifier is capable of creating a different SDVS δ^* intended for himself after receiving a valid SDVS δ. The probability that the two SDVSs are identical is at most $|\boldsymbol{G}_1|^{-1}$, i.e.,

$\Pr[\delta^* = \delta] \le |\boldsymbol{G}_1|^{-1}$.

Q.E.D.

We compare our scheme with some previous mechanisms including Kang *et al.*'s (KBD for short) [4] and the Lee *et al.*'s (LCL for short) [7] ones. Table 1 lists the detailed comparisons. From this table, it can be seen that the proposed scheme outperforms compared works in terms of computational efforts and security.

5. Conclusions

In this paper, we proposed a new efficient identity-based SDVS scheme for privacy-preserving applications. It only takes one time-consuming bilinear pairing computation for the signer and the designated verifier to separately generate and verify the SDVS. Moreover, the selective security to withstand the universal forgery attacks is formally proved in the random oracle model. We also demonstrated that our proposed scheme outperforms previously related works in terms of security and computational efforts. It is thus believed that the proposed scheme would be a better alternative for practical implementation of privacy-sensitive applications.

Table 1. Evaluation of Security and Computational Costs

Item \ Scheme	KBD	LCL	Ours
Computational cost*	5B + 8M + 3H	6B + 5M + 3H	3B + 5M + 3H
Secure against universal forgery attack	X	V	V
Secure against key-compromise attack	X	V	V

Remark*: The symbols of B, M and H separately mean the computation of one bilinear pairing, multiplication and one-way hash function.

Acknowledgement

This work was supported in part by the Ministry of Science and Technology of Republic of China under the contract number MOST 104-2221-E-019-018.

References

[1] D. Chaum and H. van Antwerpen, "Undeniable signature," *Advances in Cryptology – CRYPTO'90*, Springer-Verlag, pp. 212-216, 1990.

[2] 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, no. 1, pp. 82-93, 2008.

[3] M. Jakobsson, K. Sako and R. Impagliazzo, "Designated verifier proofs and their applications," *Advances in Cryptology – EUROCRYPT'96*, Springer-Verlag, pp. 143-154, 1996.

[4] B. Kang, C. Boyd and E. Dawson, "A novel identity-based strong designated verifier signature scheme," *The Journal of Systems and Software*, vol. 82, no. 2, pp. 270-273, 2009.

[5] K. Kumar, G. Shailaja and A. Saxena, "Identity based strong designated verifier signature scheme," *Cryptology ePrint Archive, Report 2006/134*, 2006, http://eprint.iacr.org/2006/134 (2006/4/20).

[6] J.S. Lee and J.H. Chang, "Strong designated verifier signature scheme with message recovery," *Proceedings of the 9th International Conference on Advanced Communication Technology*, vol. 1, pp. 801-803, 2007.

[7] J.S. Lee, J.H. Chang and D.H. Lee, "Forgery attacks on Kang et al.'s identity-based strong designated verifier signature scheme and its improvement with security proof," *Computers and Electrical Engineering*, vol. 36, no. 5, pp. 948-954, 2010.

[8] H.Y. Lin, T.S. Wu and Y.S. Yeh, "A DL based short strong designated verifier signature scheme with low computation," *Journal of Information Science and Engineering*, vol. 27, no. 2, pp. 451-463, 2011.

[9] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, no. 3, pp. 361-369, 2000.

[10] S. Saeednia, S. Kremer and O. Markowitch, "An efficient strong designated verifier signature scheme," *Proceedings of the 6th International Conference on Information Security and Cryptology (ICISC 2003)*, Seoul, Korea, pp. 40-54, 2003.

[11] A. Shamir, "Identity-based cryptosystems and signature schemes," *Advances in Cryptology – CRYPTO'84*, Springer-Verlag, pp. 47-53, 1984.

[12] W. Susilo, F. Zhang and Y. Mu, "Identity-based strong designated verifier signature schemes," *ACISP 2004*, LNCS 3108, pp. 313-324, 2004.

[13] G. Wang, "An Attack on not-interactive designated verifier proofs for undeniable signatures," *Cryptology ePrint archive*, 2003, http://eprint.iacr.org/2003/243 (2003/11/25).

[14] J. Zhang and J. Mao, "A novel ID-based designated verifier signature scheme," *Information Sciences*, vol. 178, no. 3, pp. 766-773, 2008.