A Lightweight ID-Based Strong Designated Verifier Signature Scheme against Key-Compromise Attacks

Han-Yu Lin*, Leo-Fan Yang and Yao-Min Hung

Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan.

* Corresponding author. Tel.: +886-2-2462-2192 ext 6656; email: lin.hanyu@msa.hinet.net

Manuscript submitted October 23, 2016; accepted May 20, 2017.
doi: 10.17706/jsw.12.8.649-656

Abstract: Lightweight cryptographic mechanisms are crucial for mobile computing as mobile devices usually have limited computing power and insufficient storage space. Some sensitive mobile applications, however, might still require simultaneously fulfilling the security requirements of confidentiality, integrity, authenticity and non-repudiation. Strong designated verifier signatures (SDVS) could be regarded as special variations of generic digital signatures, since only the intended verifier will be convinced of their authenticity. Furthermore, the intended verifier has no way to transfer his proof to any third party. In this paper, the authors incorporate the concept of timestamp with SDVS schemes to propose a lightweight and CMA-secure ID-based SDVS scheme. Our security model further allows the adversary to request the signer's private key for simulating the realistic key-compromise attack. The comparison result showed that the proposed scheme not only is more secure, but also has better computational efficiency.

Keywords: Lightweight, identity-based, strong designated verifier, digital signature, timestamp.

1. Introduction

The public key verification is always an important issue before using it. Since public keys are openly accessible to anyone, a malicious adversary might replace it with a forged one, which is referred to as public key substitution attack. To prevent such an attack, however, extra verification efforts will increase. In 1984, Shamir [1] introduced the so-called ID-based system, in which every user's public key can be explicitly verified, for that the public key is straightly his/her identifier. A system authority (SA) possessing a trapdoor secret is able to derive each user's corresponding private key with a trapdoor one-way function. Without this trapdoor secret, no one can compute the valid private key from its public one.

In public systems, there are two major mechanisms. One is the public key encryption and the other is the digital signature. The former ensures the requirement of confidentiality while the latter guarantees that of integrity, authenticity and non-repudiation. Some applications such as electronic voting [2], [3] might require the above requirements simultaneously be fulfilled. The undeniable signature scheme proposed by Chaum and Antwerpen [4] is applicable here. In their scheme, the signature verification process can only be accomplished by the verifier and the signer together. That is, the signer has the right to decide who can verify his signatures.

In a similar notion, Jakobsson *et al.* [5] addressed the designated verifier signature (DVS) scheme in which the signer also owns the right to choose verifiers for verifying his signatures. Concretely speaking, a DVS signature can be verified by anyone. However, only a designated verifier will be convinced of its authenticity due to the transcript simulation property of DVS, i.e., the intended verifier also has the ability to create a computationally indistinguishable DVS for himself. Nevertheless, their DVS scheme has some security flaws found out by Wang [6].

To allow only the designated verifier to check the signature, in 2003, Saeednia *et al.* [7] introduced the strong designated verifier signature (SDVS) scheme by combining the verifier's private key with signature verification process. Consequently, anyone without the knowledge of correct private key cannot complete the signature verification. Likewise, the designated verifier also can simulate any transcript intended for himself and hence is unable to transfer his conviction to anyone.

Considering the advantage of ID-based systems, in 2004, Susilo *et al.* [8] presented the first ID-based SDVS scheme under the hardness of Bilinear Diffie-Hellman Problem (BDHP). They first gave a generic construction which requires an additional encryption mechanism and then further proposed the other more efficient variant. In 2007, Lee and Chang [9] introduced an SDVS scheme with message recovery allowing the designated verifier to recover the original message from its signature. Two years later, Kang *et al.* [10] addressed a novel ID-based SDVS scheme with shorter signature size and lower computational costs. Unfortunately, Hsu and Lin [11] pointed out that their scheme was vulnerable to the universal forgery attack in 2014. So far, several SDVS variations [12]-[14] have been proposed.

In this paper, the authors consider lightweight cryptographic mechanisms used mobile computing environments and will propose a lightweight ID-based SDVS scheme. To satisfy the real-time application requirement, timestamps are also employed in the design of our mechanism to ensure freshness. We will show that the proposed work not only satisfies the essential security requirements of SDVS scheme, but also can resist universal forgery attacks.

2. Preliminaries

In this section, we first state the properties of bilinear pairing and then review related computational assumption utilized in the proposed scheme.

2.1. Bilinear Pairing

Let ($G1$, +) and ($G2$, ×) separately be an additive and a multiplicative group of the same prime order q and P an arbitrarily generator of G1. The notation of "aP" expresses that P added to itself a times. A mapping e: G1 × G1 → G2 is a cryptographic bilinear map which satisfies the following properties:

(i) Bilinearity:

For all $P, Q, R \in G1$ and some $a, b \in Zq^*$, we have
$e(aP, bQ) = e(P, Q)ab;$
$e(P + R, Q) = e(P, Q)e(R, Q);$
$e(P, R + Q) = e(P, R)e(P, Q);$

(ii) Non-degeneracy:

If P is a generator of $\boldsymbol{G}_1$, then $e(P, P)$ is a generator of $\boldsymbol{G}_2$, which also implies $e(P, P) \neq 1$.

(iii) Computability:

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

2.2. Bilinear Diffie-Hellman Problem; BDHP

The BDHP is, given $P, aP, bP, cP \in \boldsymbol{G}_1$ for some $a, b, c \in Z_q$, to compute $e(P, P)^{abc} \in \boldsymbol{G}_2$.

2.3. Bilinear Diffie-Hellman (BDH) Assumption

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

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

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 an advantage ε.

3. The Proposed Scheme

In this section, we introduce the proposed lightweight ID-based SDVS scheme, abbreviated to ID-SDVS. The involved parties, algorithms and construction are detailed below.

3.1 Involved Parties

An ID-SDVS scheme has three involved parties including a trusted authority (TA), a signer and a designated verifier. The TA first generates system parameters and computes each user's private key with its own master private key. The signer can produce an SDVS for his chosen message and designated verifier. After receiving the SDVS, the designated verifier can utilize his private key to verify received signature. He can also create another valid transcript for himself, but he would be unable to convince anyone of his proof.

3.2 Algorithms

An ID-SDVS 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.

– **Key Generation (KG):** The KG algorithm takes as input system parameters params, an identity ID_i and the TA's masker private key. It generates the key pair for ID_i.

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

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

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

3.3 Construction

We present a concrete construction according to previous algorithms as follows:

– **Setup:** Taking as input 1^k, the TA selects two groups (G_1, +) and (G_2, ×) of prime order q where $|q| = k$. Let P be a generator of order q over G_1 and a bilinear pairing e satisfying that $e: G_1 \times G_1 \rightarrow G_2$. There are two collision resistant hash functions H_1 and H_2 such that $H_1: \{0, 1\}^* \rightarrow G_1$ and $H_2: \{0, 1\}^* \times G_1 \times \{0, 1\}^k \rightarrow Z_q$. The public parameters params includes $\{G_1, G_2, q, P, e, H_1, H_2\}$. TA also chooses a random integer s as its master private key and then computes the corresponding public key $P_{TA} = sP$.

– **Key Generation (KG):** For each user U_i associated with the identity ID_i, TA derives his private key $S_i = sQ_i$ where $Q_i = H_1(ID_i)$ is the corresponding *public* key. Then the private key S_i is sent to the user via a secure channel.

– **SDVS-Generation (SDVS-G):** Let ID_a be a signer and ID_b be the designated verifier. To generate an *ID-SDVS* for the message $m \in_R \{0, 1\}^*$, ID_a chooses $r \in_R Z_q$ to compute

$$Y = rP_{TA}, \tag{1}$$

$$W = e(Q_b + rP, H_2(m, Y, T)S_a - P_{TA}), \text{ where T is the current timestamp.} \tag{2}$$

The *ID-SDVS* for the message m is $\delta = (T, Y, W)$ which will be delivered to the designated verifier ID_b.

- **SDVS-Verification (SDVS-V):** Upon receiving $\delta = (T, Y, W)$, ID_b first checks whether $| T' - T | \le \Delta T$ where T' is the received time and ΔT is the valid network transmission delay. To verify the ID-SDVS $\delta = (T, Y, W)$, ID_b can utilize his private key S_b to check whether

$$W = e(S_b + Y, H_2(m, Y, T)Q_a - P). \tag{3}$$

If the above quality holds, the ID-SDVS δ for m is valid. The correctness of Eq. (3) can be derived as follows. From the left-hand side of Eq. (3), we have

$$\begin{aligned}
&W \\
&= e(Q_b + rP, H_2(m, Y, T)S_a - P_{TA}) && \text{(by Eq. (2))} \\
&= e(Q_b + rP, s(H_2(m, Y, T)Q_a - P)) = e(Q_b + rP, H_2(m, Y, T)Q_a - P)^s = e(s(Q_b + rP), H_2(m, Y, T)Q_a - P) \\
&= e(sQ_b + rP_{TA}, H_2(m, Y, T)Q_a - P) && \text{(by Eq. (1))} \\
&\quad = e(S_b + Y, H_2(m, Y, T)Q_a - P)
\end{aligned}$$

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

- **Transcript-Simulation (TS):** ID_b is able to produce a valid ID-SDVS for his chosen message m' with his private key. First, he chooses $Y' \in_R G_1$ and then computes

$$W' = e(S_b + Y', H_2(m, Y', T)Q_a - P), \text{ where } Y' \text{ is the current timestamp.} \tag{4}$$

It is obvious that $\delta' = (T', Y', W')$ is another valid ID-SDVS for m'.

4. Security and Efficiency

We first analyze the security of our scheme and then evaluate its efficiency in terms of computational costs.

Theorem 1. The proposed lightweight ID-SDVS scheme is $(t, qH1, qH2, qKG, qG, \varepsilon)$-secure against universal forgery attacks under adaptive chosen-message attacks (EF-CMA) in the random oracle model if there is no probabilistic polynomial-time adversary A that can break the BDHP with non-negligible advantage.

Proof: We first assume that there is a probabilistic polynomial-time adversary A who has the non-negligible advantage to forge a valid ID-SDVS in the proposed scheme. The adversary A is allowed to ask at most q_{H_i} H_i random oracles (for $i = 1$ and 2), q_{KG} *KG* and q_G *SDVS-G* queries. Based on the adversary A' forging ability, we will show how to construct a BDHP solving algorithm B that utilizes A as its subroutine. Let the BDHP instance taking by B is (P, aP, bP, cP) and it will output the value $e(P, P)^{abc}$ with a non-negligible probability. In this proof, B also simulates a challenger to A in the following game.

Setup: The challenger B first initializes the Setup(1^k) algorithm, sets $P_{TA} = aP$ and gets public parameters $params = \{G_1, G_2, q, P, e\}$. It then prepares a random tape RT composed of a long sequence of random bits and simulates two runs of the proposed ID-SDVS scheme to the adversary *A* by taking *(params, P_{TA}, RT)* as inputs.

Phase 1: *A* can issue the following queries adaptively:

– *$H_1(ID_i)$ oracle:* When *A* queries *$H_1(ID_i)$* oracle, *B* first seeks the stored *H_1*_table for a matched entry. Otherwise, *B* chooses $v_1 \in_R Z_q$, inserts the entry *(ID_i, v_1, v_1P)* into *H_1*_table and returns v_1P as a result. Note that in the *j*-th and *(j* + 1)-th queries, *B* directly returns *bP* and *cP.*

– *$H_2(m, Y, T)$ oracle:* Once *A* issues an *$H_2(m, Y, T)$* oracle, *B* will check the maintained *H_2*_table for a matched entry. Else, *B* chooses $v_2 \in R\ Z_q$, stores the entry *(m, Y, T, v_2)* into *H_2*_table and sends v_2 back as a result.

– *KG(ID_i) oracle:* If *A* asks an *KG(ID_i)* oracle for $i \neq j$ and $i \neq j + 1$, *B* first runs the corresponding *$H_1(ID_i)$* oracle to obtain v_1 and then computes the private key as *v_1aP.* Finally, *B* stores *(ID_i, v_1aP, v_1P)* into maintained *KG*_table and sends the private-public key pair *(v_1aP, v_1P)* back as a result.

– *SDVS-G(m, ID_i, ID_v) oracle:* If *A* asks an *SDVS-G(m, ID_i, ID_v)* oracle for $i \neq j$ and $i \neq j + 1$, *B* first runs the *KG(ID_i)* oracle, obtains the private key S_i and then computes a valid ID-SDVS $\delta = (T, Y, W)$ according to the SDVS-G algorithm. In case that *i = j or i = j + 1, B* can first run the *KG(ID_v)* oracle, get the private key S_v and then derive a valid ID-SDVS $\delta = (T, Y, W)$ according to the TS algorithm. At last, the generated ID-SDVS $\delta = (T, Y, W)$ is returned as a result. This query might abort only when an SDVS-G*(m, ID_i, ID_v)* oracle for *(i = j, v = j + 1) or (i = j + 1, v = j)* is made.

Forgery: After making sufficient queries, *A* will generate a forged ID-SDVS $\delta^* = (T^*, Y^*, W^*)$ for his arbitrarily chosen identities *(ID_a*, ID_b*)* and message *m*.* Note that δ^* cannot be outputted by any *SDVS-G* oracle and *KG(ID_i)* queries for *(i = a*)* and *(i = b*)* have never been asked.

Analysis of the game: Since the adversary *A* is supplied with an identical random tape during the simulated two rounds, it will query the same sequence of oracles. The challenger *B* always returns answers as those in the round one, except for the final *$H_2(m^*, Y^*, T^*)$* oracle with respect to the resulted ID-SDVS $\delta^* = (T^*, Y^*, W^*)$. In order to obtain another valid ID-SDVS, at this time, *B* returns a new value $v_2^{**} \in_R Z_q$ instead of original v_2^*. According to the "Forking lemma" [15], when *A* eventually produces a valid forgery $\delta^{**} = (T^*, Y^*, W^{**})$ where $H_2(m^*, Y^*, T^*) = \ \ v_2^{**}$ and $(ID_a^* = ID_j, ID_b^* = ID_{j+1})$, *B* could obtain

$$\begin{aligned} W^* &= e(S_b + Y^*, v_2^*Q_a - P) = e(S_b, v_2^*Q_a - P)e(Y^*, v_2^*Q_a - P) \\ &= [e(S_b, v_2^*Qa)/e(S_b, P)][e(Y^*, v_2^*Q_a)/e(Y^*, P)], \end{aligned} \tag{4}$$

and

$$\begin{aligned} W^{**} &= e(S_b + Y^*, v_2^{**}Q_a - P) = e(S_b, v_2^{**}Q_a - P)e(Y^*, v_2^{**}Q_a - P) \\ &= [e(S_b, v_2^{**}Q_a)/e(S_b, P)][e(Y^*, v_2^{**}Q_a)/e(Y^*, P)]. \end{aligned} \tag{5}$$

Combining Eqs. (4) and (5), we can further derive

$$\begin{aligned} & W^*/W^{**} = e(S_b + Y^*, (v_2^* - v_2^{**})Q_a) = e(S_b, (v_2^* - v_2^{**})Q_a)e(Y^*, (v_2^* - v_2^{**})Q_a) \\ \Rightarrow\ & (W^*/W^{**}/e(Y^*, (v_2^* - v_2^{**})Q_a) = e(Sb, (v_2^* - v_2^{**})Q_a) = e(acP, (v_2^* - v_2^{**})bP) \\ & = e(P, (v2^* - v2^{**})P)abc \end{aligned}$$

Therefore, *B* could solve the BDHP by computing.

$$e(P, P)^{abc} = [W * / W * * / e(Y*, (v_2 * - v_2 * *)Q_a)]^{(v_2 * - v_2 * *)^{-1}}$$

Q.E.D.

Theorem 2. The proposed lightweight ID-SDVS scheme ensures signer ambiguity even under the key-compromise attack.

Proof: Signer ambiguity means that it should be computationally indistinguishable for anyone to identify the real signer with respect to an ID-SDVS from only two candidates. The key-compromise attack further allows the adversary to obtain the compromised private key. In the proposed scheme, the verification equality, Eq. (3), can be further expressed as $W = e(S_b + Y, H_2(m, Y, T)Q_a - P) = e(Q_b + rP, H_2(m, Y, T)S_a - P_{TA})$. From the above equality, one can see that even if an adversary gets the signer's private key, he is still unable to perform the signature verification equality due to the random number r. Consequently, we can claim that the proposed protocol satisfies the requirement of signer ambiguity even under the key-compromise attack.

Q.E.D.

Theorem 3. The proposed lightweight ID-SDVS scheme fulfills the security requirement of non-transferability.

Proof: The non-transferability requirement guarantees that a designated verifier cannot transfer his conviction to any third party. According to the Transcript-Simulation (TS) phase of our construction, a designated verifier is capable of generating another valid ID-SDVS δ^* intended for himself with his private key. Hence, the non-transferability property is satisfied in the proposed mechanism.

Q.E.D.

To demonstrate the security and computational performance of our mechanism, we make a comparison with some existing schemes including Kang *et al.'s* (KBD for short) [10] and Lee *et al.'s* (LCL for short) [16] ones. To simplify the evaluation of computational performance, we just consider the most time-consuming operation, i.e., bilinear pairing computation. Let T_B be the time for computing one bilinear pairing operation. The detailed comparisons are listed as Table I. From the table, one can observe that both the LCL and our schemes are secure against universal forgery and key-compromise attacks. Nevertheless, the LCL scheme requires the highest computational efforts, i.e., $6\ T_B$. Therefore, we claim that the proposed scheme would be a secure and efficient alternative for practical applications.

Table 1. Comparisons of Security and Computational Efficiency

Scheme Item	KBD	LCL	Ours
Resist Universal Forgery Attack	X	O	O
Resist Key-Compromise Attack	X	O	O
Computational cost for SDVS-G	$2T_B$	$2T_B$	T_B
Computational cost for SDVS-V	T_B	$2T_B$	T_B
Computational cost for TS	$2T_B$	$2T_B$	T_B
Computational cost for entire scheme	$5T_B$	$6T_B$	$3T_B$

5. Conclusions

For facilitating the requirement of mobile computing, in this paper, the authors employed the concept of timestamp to propose a lightweight ID-based strong designated verifier signature scheme. Based on the intractable bilinear Diffie-Hellman problem, we formally proved that our scheme achieves the EF-CMA security even under the key-compromise attack. We also analyzed the essential security requirements of our ID-SDVS scheme. To ensure the feasibility of our work, we compared it with previous related mechanisms and the results reveal better security and lower computational cost of our protocol. More specifically, each phase of our construction only takes one time-consuming bilinear pair

Acknowledgement

This work was supported in part by the Ministry of Science and Technology of Republic of China under the contract number MOST 105-2221-E-019-063.

References

[1] Shamir, A. (1984). Identity-based cryptosystems and signature schemes. *Advances in Cryptology – CRYPTO'84: Springer-Verlag,* 47-53.

[2] Ray, I., & Narasimhamurthi, N. (2001). An anonymous electronic voting protocol for voting over the Internet. *Proceedings of the 3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS'01)* (pp. 188-190), California.

[3] Schoenmakers, B. (1999). A simple publicly verifiable secret sharing scheme and its application to electronic voting. *Advances in Cryptology – CRYPTO'99: Springer-Verlag,* 148-164.

[4] Chaum, D., & Van, A. H. (1990). Undeniable signature. *Advances in Cryptology*.

[5] Jakobsson, M., Sako, K., & Impagliazzo, R. (1996). Designated verifier proofs and their applications. *Advances in Cryptology – EUROCRYPT'96: Springer-Verlag,* 143-154.

[6] Wang, G. (2003). An Attack on not-interactive designated verifier proofs for undeniable signatures, Cryptology ePrint archive.

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

[8] Susilo, W., Zhang, F., & Mu, Y. (2004). Identity-based strong designated verifier signature schemes.

[9] Lee, J. S., & Chang, J. H. (2007). Strong designated verifier signature scheme with message recovery. *Proceedings of the 9th International Conference on Advanced Communication Technology*.

[10] Kang, B., Boyd, C., & Dawson, E. (2009). A novel identity-based strong designated verifier signature scheme. *The Journal of Systems and Software, 82(2)*, 270-273.

[11] Hsu, H., & Lin, H. Y. (2014). Universal forgery attack on a strong designated verifier signature scheme. *The International Arab Journal of Information Technology*, *11(5)*, 425-428.

[12] Huang, X., Susilo, W., Mu, Y., & Zhang, F. (2008). Short designated verifier signature scheme and its identity-based variant. *International Journal of Network Security*, *6(1)*, 82-93.

[13] Lin, H. Y., Wu, T. S., & Yeh, Y. S., (2011). A DL based short strong designated verifier signature scheme with low computation. *Journal of Information Science and Engineering*, *27(2)*, 451-463.

[14] Zhang, J., & Mao, J. (2008). A novel ID-based designated verifier signature scheme. *Information Sciences, 178(3)*, 766-773.

[15] Pointcheval, D., & Stern, J. (2000). Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, *13*, 361-369.

[16] Lee, J. S., Chang, J. H., & Lee, D. H. (2010). Forgery attacks on Kang *et al.*'s identity-based strong designated verifier signature scheme and its improvement with security proof. *Computers and Electrical Engineering*, *36(5)*, 948-954.

Han-Yu Lin received 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 Ph.D. degree in computer science and engineering from the National Chiao Tung University, Taiwan in 2010. He has been an Assistant Professor in the Department of

Computer Science and Engineering of National Taiwan Ocean University since August 2012. His research interests include Cryptology, Network Security, Digital Forensics, Cloud Computing Security and E-commerce Security.

Leo-Fan Yang received his BS degree in computer science and information engineering from Chinese Culture University, Taiwan in 2015. Now he is a graduate student in the department of computer science and engineering of National Taiwan Ocean University, Taiwan. His research interests lie in Cryptography, Digital Watermarking and Information Security.

Yao-Ming Hung received his BS degree in computer science and information engineering from Chinese Culture University, Taiwan in 2014. Now he is a graduate student in the department of computer science and engineering of National Taiwan Ocean University, Taiwan. His research interests lie in Cryptography and Information Security.