An Improved Designated Verifier Signature Schemes Using Bilinear Paring

Q.Z. Yue, H. Shen, J.H. Chen
School of Mathematics and Statistics
Wuhan University
Wuhan, China

***Abstract*—In a designated verifier signature scheme, only the designated person can verify the validity of the signature. Due to such attribute, it could be used in many fields such as financial payment system, e-voting, and e-taxation. Recently, Lee et al. proposed an identity based signature scheme to overcome weaknesses in previous schemes. In this paper, we will point out that Lee et al.'s scheme suffers from two kinds of attacks. To enhance security, an improved scheme will be also proposed. Security analysis shows that that the proposed scheme is provably secure in the random oracle model and could overcome weaknesses in Lee et al.'s scheme. Performance analysis demonstrates that our scheme could overcome weaknesses in Lee et al.'s weaknesses at the cost of increasing computational cost slightly.**

Keywords-designated verifier signature; bilinear pairing; diffie-hellman problem; security model

I. INTRODUCTION

With the development of the Internet, digital signature is widely used in many fields, So we need to change something to adapt to the special environment conditions. In some scenarios such as e-voting[1,2], they want that only the system can verify the valid of their signature. For this reason, normal digital signature which anyone can know the original signer's attitude is not suitable.

To solve this problem, Jakobsson et al. introduced the concept of designated verifier signature (DVS) in 1996[3]. This ensures that any third party cannot get any useful knowledge of the signed information, and the signer cannot deny the fact that he has signed the message when the signature is valid.

In order to improve the security and efficiency of DVS scheme, Saeednia et al.[4] formalized the strong DVS notation and proposed a novel scheme in 2003. Many kinds of new DVS schemes were proposed [5,6,7,8,9] following that. In 2009, Kang et al.[10] proposed a scheme with new construction and Yoon gave another secure and effective scheme in 2011. Unfortunately, Lee et al. point out that Yoon's scheme is vulnerable to replay-attack. It is easy to be forged for the third part when it intercepts the signature through controlling the communication channel.

II. PRELIMINARIES

In this section, we briefly introduce some knowledge about bilinear pairing and some necessary security notions

A. Bilinear Pairing

Let $(G_1,+)$ and $(G_2,\times)$ be two cyclic groups over an elliptic curve which have the same prime order q. And P denotes the generator of G_1. A bilinear map: $e: G_1 \times G_1 \to G_2$ satisfies the following properties:

1. Bilinear: $e(P_1+P_2,Q)=e(P_1,Q)e(P_2,Q)$, $\forall P_1,P_2,Q \in G_1$

$e(P,Q_1+Q_2)=e(P,Q_1)e(P,Q_2)$, $\forall P,Q_1,Q_2 \in G_1$

2. Non-degeneracy: There exist $P,Q \in G_1$ such that $e(P,Q)\neq 1$.

3. Computability: There is a polynomial-time algorithm to compute for all.

Definition1. Bilinear Diffie-Hellman Problem (BDHP) is to compute the value of $e(P,P)^{xyz}$ when given xP, yP $zP \in G_1$, for unknown $x,y,z \in Z_q$.

Definition2. Bilinear Diffie-Hellman(BDH) assumption states that in probabilistic polynomial time , there is no algorithms which can solve BDHP with non-negligible advantage.

B. Security Model

Before introducing the security model of our scheme, some notations are defined in as follows.

- C: an adversary who can forge the signature with an unnegligible advantage;
- Λ: an algorithm which simulates with the adversary C to solve BDHP;
- H_1 query: the adversary can query the value of $H_1(\cdot)$ and the algorithm Λ responds with the public key of signer or the verifier;
- H_2 query: the adversary can query the value of $H_2(\cdot)$ and the algorithm Λ responds with a point on the elliptic curve;
- H_3 query: the adversary can query the value of $H_3(\cdot)$ and the algorithm Λ responds with any values as he chooses;

●H2_list: a list to store the input and output values that the adversary has queried in the $H_2(\cdot)$ query;

●H3_list: a list to store the input and output values that the adversary has queried in the $H_3(\cdot)$ query;

If there is an adversary C can forge the signature with non-negligible advantage, we could construct an algorithm Λ which can solve BDHP through the following game.

Initially, the adversary C gets the system parameter.

C could make $H_1(\cdot)$ query, $H_2(\cdot)$ query and $H_3(\cdot)$ query. The algorithm Λ answers C 's queries by using any reasonable data instead of the signer. When C queries secret information including the users' private key, the game aborts and fails.

We could solve BDHP through the following game by two signatures which are generated by the adversary using the same message.

III. Review of Lee et al.'s scheme

In the Scheme of Lee et al,We assume that there are two candidates in the scheme: the signer Alice and the verifier Bob. The details are described in the following.

A. Setup Phase

In this phase, the PKG (private key generator centre) has to choose two cyclic groups: $(G_1,+)$ and $(G_2,\times)$, where q is the common order of the two groups and define the bilinear paring map as $\hat{e}: G_1 \times G_1 \to G_2$. Then the PKG generates the master secret key $S \in Zq^*$ and two one-way hash functions: $H_1(\cdot): \{0,1\}^* \to G_1$, $H_2(\cdot): \{0,1\}^* \to Z_q^*$ as its system parameters. At last, the PKG publishes the system public parameters $\{\hat{e}, G_1, G_2, q, H_1(\cdot), H_2(\cdot)\}$.

B. Key-Extract Phase

Everyone has a unique ID which can be used to calculate their public key $H_1(ID)$. Taking the user's ID as input, the PKG outputs the user's private key $S_{ID} = s \cdot H_1(ID)$ and sends it to the user though the secure channel.

C. Sign Phase

When Alice wants to make a signature,he has to compute Bob's public key $Q_{ID_B} = H_1(ID_B)$. Then, Alice chooses a correct timestamp T and computes $r = H_2(T)$ Alice computes $\delta = xQ_{ID_A}$ and $\sigma = H_2(M, \hat{e}(xQ_{ID_B}), rS_{ID_A}))$,Finally, Alice sends $\{M, T, \delta, \sigma\}$ to Bob as a signature.

D. Verify Phase

When Bob receives the information $\{M, T, \delta, \sigma\}$ from Alice, he first checks the timestamp T. If T does not meet the requirement, Bob ignores the information. Otherwise, Bob computes $r = H_2(T)$. After that, Bob checks the equation as follows: $\sigma \stackrel{?}{=} H_2(M, \hat{e}(rS_{ID_B}, \delta))$.

If the equation is hold, Bob considers the information as a valid signature; otherwise, Bob ignores it.

E. Transcript Simulation Phase

After accepting the message and signature , Bob produces the transcripts. Bob selects a random number $r' \in Z_q^*$, which is different from r , and then computes $\sigma' = H_2(M, \hat{e}(r'S_{ID_B}, \delta))$. Finally, Bob stores $\{r', \sigma', \delta\}$.

IV. Cryptanalysis of Lee's scheme

A. Attacks On Lee Et Al.'S Scheme

In this section, we will analyze the weaknesses of Lee et al.'s scheme. We show two kinds of attacks as follows.

(1) We assume that C is an adversary with an identity ID_C . So we can calculate his public key $Q_{ID_C} = H_1(ID_C)$ and private key $S_{ID_C} = s \cdot H_1(ID_C)$. His goal is to forge Alice's signature. The details are shown in the followings where M^* is the message that C wants to forge.

Firstly, C gets the Alice's signature $\{M, T, \delta, \sigma\}$ through controling the communication channel between Alice and Bob. Then C chooses a new timestamp T^* and random number x^* to calculate $r^* = H_2(T^*)$ and $\delta^* = x^* Q_{ID_C}$. After that, the adversary calculates the signature $\sigma^* = H_2(M^*, \hat{e}(x^* Q_{ID_B}, r^* S_{ID_C}))$. At last C sends the information $\{M^*, T^*, \delta^*, \sigma^*\}$ to Bob.

When Bob receives the information , he will check the valid of the signature. By the reason that the adversary chooses another timestamp T^* which can be suitable to the current time, it could be accepted by Bob. Then Bob calculates $r^* = H_2(T^*)$ and $H_2(M^*, \hat{e}(r^* S_{ID_B}, \delta^*))$. By the following equation, we will show the signature is valid.

$$H_2(M^*, \hat{e}(r^* S_{ID_B}, \delta^*)) = H_2(M^*, \hat{e}(x^* Q_{ID_B}, r^* S_{ID_C})) = \sigma^*$$

(2) When the verifier Bob is the adversary, he can forge every signature that he is the designated verifier. We assume M^* that is the message that Bob wants to forge. Bob can get T^* and δ^* from any signature created by Alice. He just makes $T^* = T$ and $\delta^* = \delta$. Bob computes $\sigma^* = H_2(M^*, \hat{e}(r^* S_{ID_B}, \delta^*))$. It is easy to see that $\{M^*, T^*, \delta^*, \sigma^*\}$ can be instead of $\{M, T, \delta, \sigma\}$. And

Alice cannot deny that $\{M^*, T^*, \delta^*, \sigma^*\}$ is not created by himself.

B. The Unsecure Reason of Lee'S Scheme

By analyzing Lee's scheme, we find that the weakest point of the scheme. We get equation $\hat{e}(xQ_{ID_B}, rS_{ID_A}) = \hat{e}(xQ_{ID_A}, rS_{ID_B}) = \hat{e}(\delta, rS_{ID_B})$. So when the adversary intercepts one signature created by Alice, he can forge any signatures. What is more, the timestamp T is useless and can be instead easily in the scheme.

V. OUR PROPOSED SCHEME

A. Description of Our Proposed Scheme

In this subsection, we will construct a new scheme. Like Lee's scheme, our scheme is also composed of five phases. Alice and Bob are the signer and the verifier, respectively.

(1) Setup phase

Firstly the PKG has to choose two cyclic groups: $(G_1, +)$ and $(G_2, \times)$, where q is the common order of the two groups and define the bilinear paring map as $e: G_1 \times G_1 \to G_2$.Then it generates the system parameters including master secret key $s \in Z_q^*$ and three one-way hash functions $H_1(\cdot)$, $H_2(\cdot)$, $H_3(\cdot)$. The definition of hash functions are : $H_1(\cdot): \{0,1\}^* \to G_1$, $H_2(\cdot): \{0,1\}^* \to G_1$, $H_3(\cdot): \{0,1\}^* \cdot G_2 \cdot G_1 \to Z_q^*$. The public key of the PKG is $P_{pub} = sP$ where P is a generator in.

(2) Key-Extract phase

Everyone has a unique ID which can be used to calculate their public key. Taking the user's ID as input, the PKG outputs the user's private key $s_{ID} = s \cdot H_1(ID)$ and sends it to the user though the secure channel.

(3) Sign phase

In this phase, Alice makes a signature of the message M. Firstly, he computes $\rho = r \cdot Q_{ID_A}$ where $r \in Z_q$ is a random number. Next, Alice gets $T = e(H_2(M), S_{ID_A})$ and $W = e((H_3(M,T,\rho)+r)Q_{ID_B}, S_{ID_A})$ .Finally,Alice sends the signature information $\{M, \rho, T, W\}$ to Bob.

(4) Verify phase

When Bob receives the information $\{M, \rho, T, W\}$ from Alice, he first computes the value of $H_3(M,T,\rho)$. After that, Bob checks the equation $W \stackrel{?}{=} e(H_3(M,T,\rho)Q_{ID_A} + \rho, S_{ID_B})$. If it is hold, Bob accepts the information and considers it as a valid signature.

(5) Transcript simulation phase

When Bob verifies and accepts the information from Alice, he selects a random $\rho' \in G$, and computes $W = e(H_3(M,T,\rho')Q_{ID_A} + \rho', S_{ID_B})$. Finally, Bob stores $\{\rho', T, W\}$.

B. Security Analysis of The Proposed Scheme

In this subsection, we will do security analysis of our scheme.

(1) Correctness. In our schemes, the signature $W = e((H_3(M,T,\rho)+r)Q_{ID_B}, S_{ID_A})$ is created by Alice, and Bob can check the valid of the signature by the followings.

$W = e(H_3(M,T,\rho)Q_{ID_A} + \rho, S_{ID_B}) = e((H_3(M,T,\rho)+r)Q_{ID_B}, S_{ID_A})$.

(2) In order to make sure that only Bob can verify the valid of the signature, Alice should use Bob's public key in the signing process. At the same time, to guarantee Bob to verify the signature successfully, he should make use of his private key and Alice's public key.

(3) In our scheme, we add T in the signature which is not the timestamp but a value of bilinear paring. If Bob wants to forge a signature from Alice, he has to create a new T^*. Because Alice uses his private key to generate $T = e(H_2(M), S_{ID_A})$, it is difficult for Bob to generate T^* without knowing the private key of Alice. When Alice wants to check whether the signature is created by him, he just needs the value of M, ρ to compute the value of T.

Now, we will show the secure proof of our proposed scheme.

Theorem1. If an adversary C can forge a valid signature where the signer is Alice and the designated verifier is Bob, then there exists an algorithm Λ which can solve the BDHP in a polynomial time with an unnegligible advantage.

Proof: In our proof, there are three participators including the signer Alice, the verifier Bob and the adversary C. Algorithm Λ will replace the adversary C's interaction with the signer by simulation. In the simulation, the algorithm Λ can provide any information that the adversary queries except the signature on the message m.

Initially, the adversary C gets $\{e, G_1, G_2, P, q, H_1(\cdot), H_2(\cdot), H_3(\cdot), P_{pub}\}$ for the KGC. As $H_1(\cdot)$ is a map to point hash operation, the user's public key can be shown by point multiplication on the elliptic curve. Without losing generality, we regard Alice's public key as $Q_{ID_A} = aP$ and Bob's public key as $Q_{ID_B} = bP$. Now the purpose of the algorithm is to solve the BDHP by calculating $e(P,P)^{abs}$ without knowing the values $a, b, s \in Z_q$. In the simulation, Λ will inject (aP, bP, sP) to the communication with the adversary C. When C queries the private key for

either user in the games, Λ aborts and fails. Finally, C will generate two valid signature, but for the same message m and the same random number r. We can solve the BDHP when C forges two valid signatures without failing and aborting. More concretely, the game between Λ and C is described as follows, where H2_list is a list to store $(m, H_2(m))$ and H3_list is a list to store $(m, T, \rho, H_3(m, T, \rho))$.First round:

H_1 query: When C requests the value of $H_1(ID)$, Λ will responds with the list $\{(ID_A, aP), (ID_B, bP)\}$. H_2 query: When C requests the value of $H_2(m)$, Λ will search the H2_list first. Otherwise, Λ chooses a random point $t \in G_1$ and return it. Λ adds the value (m, t) into the H2_list. H_3 query: When C requests the value of $H_3(m, T, \rho)$, Λ will choose a random number $k_1 \in Z_q$ and return it. Λ adds the value (m, T, ρ, k_1) into the H3_list. Finally, the adversary C generates the valid signature $\{M, \rho, T, W_1\}$.

Second round: H_1 query: When C requests the value of $H_1(ID)$, Λ will responds with the list $\{(ID_A, aP), (ID_B, bP)\}$. H_2 query: When C requests the value of $H_2(m)$, Λ will search the H2_list first. Otherwise, Λ chooses a random number $t \in G_1$ and return it. At the same time, Λ adds the value (m, t) into the H2_list. H_3 query: When C requests the value of $H_3(m, T, \rho)$, Λ will search the H3_list first. If it has been queried, Λ will choose a number $k_2 \in Z_q$ where $k_1 - k_2 \equiv 1 (\mathrm{mod}\, q)$ and return it. Finally, the adversary C generates the valid signature $\{M, \rho, T, W_2\}$.

After obtaining $\{M, \rho, T, W_1\}$ and $\{M, \rho, T, W_2\}$, Λ can compute $d = \frac{W_1}{W_2}$ as the solution of the BDHP. The detail calculating process is as follows.

$$d = \frac{W_1}{W_2} = \frac{e\left((k_1 + r) Q_{ID_B}, S_{ID_A}\right)}{e\left((k_2 + r) Q_{ID_B}, S_{ID_A}\right)} = e\left((k_1 - k_2) Q_{ID_B}, S_{ID_A}\right) = e\left(Q_{ID_B}, S_{ID_A}\right) = e(bP, saP) = e(P, P)^{abs}$$

C. Efficiency Analysis

In this subsection, we will give a performance comparison between our scheme and the related DVS schemes which need to use bilinear pairing. The main calculates in the schemes include pairing operation T_{pair}, point multiplication over an elliptic curve T_{mul}, and MTP(map to point) hash operation T_{mtp}. The comparison results shows in Table 1 as follows.

TABLE II.THE COMPARISON RESULTS OF EFFICIENCY.

	Signing cost	Verifying cost
Lee's scheme	$3T_{mul} + 1T_{mtp} + 1T_{p}$	$1T_{mul} + 1T_{par}$
Our proposed scheme	$3T_{mul} + 2T_{mtp} + 2T$	$1T_{mul} + 1T_{par}$

From Table 1, we could get the computational cost of our scheme is slightly higher than that of Lee et al.'s scheme. It is well known that the security is the first important for cryptographic scheme. Therefore, it is acceptable to enhance security at the cost of increasing computational cost slightly.

VI. CONCLUSIONS

In this paper, we analyze the weaknesses of Lee et al.'s scheme carefully. After that, we propose two kinds of attacks against their scheme. Finally, we construct a new DVS scheme which can withstand those two attacks. Although our scheme seems less efficient, it processes higher security level.

REFERENCES

[1] Cheng-Chi Lee, Yan-Ming Lai, Chin-Ling Chen, Lung Albert Chen. A Novel Designated Verifier Signature Scheme Based on Bilinear Pairing, 2013, Vol.42, No.3.

[2] E.-J. Yoon. An efficient and secure identity-based strong designated verifier signature scheme. Information Technology and Control, 2011, Vol. 40, No. 4, 323-329.

[3] M. Jakobsson, K. Sako, R. Impagliazzo. Designated verifier proofs and their applications. Lecture Notes in Computer Science, 1996, Vol. 1070/1996, pp. 143-154.

[4] S. Saeednia, S. Kremer, O. Markowitch. An efficient strong designated verifier signature scheme. In: Lecture Notes in Computer Science, 2004, Vol. 2971/2004, 40-54.

[5] J. Zhang, J. Mao. A novel ID-based designated verifier signature scheme. Information Sciences, 2008, Vol. 178, No. 3, 766-773.

[6] F. Laguillaumie, D. Vergnaud. Multi-designated ve-rifiers signatures: anonymity without encryption. In: Information Processing Letters, 2007, Vol. 102, No. 2-3, pp. 127-132.

[7] C. Y. Ny, W. Susilo, Y. Mu. Universal designated multi-verifiers signature schemes. In: Proceedings of ICPADS'05, 2005, pp. 305-309.

[8] G. Shailaja, K. P. Kumar, A. Saxenh. Universal designated multi-verifier signature without random oracles. In: Proceedings of ICIT'06, 2006, pp. 168-171.

[9] Y. Ming, Y. Wang. Universal designated multi-verifier's signature scheme without random oracles. Wuhan University Journal of Natural Sciences, 2008, Vol. 13, No. 6, 685-691.

[10] B. Kang, C. Boyd, E. Dawson. Identity-based strong designated verifier signature schemes: attacks and new construction. Computers and Electrical Engineering, 2009, Vol. 35, No. 1, 49-53.