Attack on Kang et al.'s Identity-Based Strong Designated Verifier Signature Scheme

Hongzhen Du[1,2] and Qiaoyan Wen[1]

1 School of Science, Beijing University of Posts and Telecommunications,
Beijing 100876, China
2 Mathematics Department, Baoji University of Arts and Sciences,
Baoji 721007, China
E-mail: duhongzhen@gmail.com

Abstract: In this paper, we present a universal forgery attack on Kang et al.'s identity-based strong designated verifier signature (IBSDVS) scheme. We show anyone can forge a valid IBSDVS on an arbitrary message without the knowledge of the private key of either the signer or the designated verifier. Moreover, we point out that Kang et al.'s scheme does not satisfy the properties of strongness and non-delegatability. At last, an improved IBSDVS scheme for Kang et al.'s scheme is presented, and it is provably secure and achieves all the requirements for an IBSDVS.

Keywords: designated verifier signature, bilinear pairings, cryptanalysis

1 Introduction

The concept of designated verifier signature (DVS) was first proposed by Jakobsson et al. [1] at Eurocrypt'96. Such signatures provide message authentication without non-repudiation and have the property that only designated recipient can check their validity. Designated verifier signatures have several applications such as E-voting, call for tenders and software licensing. In [1], Jakobsson et al. also introduced a stronger version of DVS called strong designated verifier signature (SDVS). In this stronger scheme, no third party can verify the validity of a designated verifier signature, since the designated verifier's private key is required in the verification phase. A SDVS scheme should satisfy the properties of strongness, unforgeability, non-transferability privacy, source hiding, and non-delegatability. In 2003, Saeednia et al. [2] first formalized the notion of the SDVS and proposed an efficient scheme in their paper. In 2004, Laguillaumie et al. constructed two DVS schemes in [3, 4]. In [5], Susilo et al. first presented an identity-based strong designated verifier signature (IBSDVS) scheme which is an identity-based variant of SDVS scheme. Thereafter, several IBSDVS schemes [6, 7, 8] have been proposed.

Recently, Kang et al. [9] proposed a new IBSDVS scheme which is more efficient than previous schemes [5, 6, 7]. Kang et al. claimed that their scheme is strong and satisfies the security property of unforgeability. However, we point out that their scheme is universally forgeable and it can not satisfy the property of strongness. Moreover, we find that their scheme is vulnerable to a delegatability attack. That is, in their scheme, a signer can delegate his signing ability, with respect to a fixed designated verifier, to a third party without disclosing his private key.

2 Review of Kang et al.'s ID-based Strong Designated Verifier Signature Scheme

We first review Kang et al.'s IBSDVS scheme [9] in brief.

- **Setup**: A bilinear map e: $G_1 \times G_1 \rightarrow G_2$, for G_1 and G_2 are groups of same prime order q. And P is a generator of group G_1. Then, a Private Key

Generation (PKG) centre picks a random $s \in Z_q{}^*$ as the master key and computes the corresponding public key $P_{pub}=sP$. H_1 and H_2 are cryptographic hash functions such that H_1: $\{0, 1\}^* \rightarrow G_1$ and H_2: $\{0, 1\}^* \rightarrow Z_q{}^*$. The system parameters are $params= <q, G_1, G_2, e, P, P_{pub}, H_1, H_2>$.

- **Key-Extract**: Given a user's identity ID, PKG computes $Q_{ID}=H_1(ID)$ and outputs the user's private key $d_{ID}=sQ_{ID}$.

Assume that Alice is the signer and Bob is the designated verifier, and Alice and Bob have their private/public key pairs (d_A, Q_A) and (d_B, Q_B), respectively.

- **IBSDVS- Sign**: To sign a message m for Bob, Alice performs as below.

1. Choose a random value $k \in Z_q{}^*$ and compute $t=e(P, Q_B)^k$.
2. Set $h=H_2(m, t)$.
3. Compute $T=kP+hd_A$ and $\sigma=e(T, Q_B)$.

The signature on the message m is (t, σ).

- **IBSDVS-Verify**: Given $params$, the signer's public key Q_A and the signature (t, σ) on m, Bob sets $h=H_2(m, t)$ and accepts the signature if and only if the following equation holds.

$$\sigma = te(Q_A, d_B)^h$$

- **IBSDVS-Simulation**: Bob can produce the signature (t, σ) intended for himself by performing the following:

1. Choose a random value $k' \in Z_q{}^*$ and compute $t' = e(P, Q_B)^{k'}$.
2. Set $h' = H_2(m, t')$.
3. Compute $\sigma' = t'e(Q_A, d_B)^{h'}$.

Then, the tuple (t', σ') is a valid signature on message m.

About the correctness and the security analysis of the scheme refer to [9].

3 Cryptanalysis of Kang et al.'s IBSDVS Scheme

In this section, we show that Kang et al.'s scheme can not achieve the requirements of unforgeability, strongness and non-delegatability.

3.1 Forgery Attack

We propose a universal forgery attack on Kang et al.'s scheme.

Assume that Charlie is an adversary without the knowledge of the private keys of the signer Alice and the designated verifier Bob. But he can forge a valid IBSDVS on any message with his choice as follows:

After intercepting a designated verifier signature (t, σ) on a message m (It is easy), Charlie performs as below:

1. Set h=H_2 (m, t).
2. Compute v=h^{-1}(mod q).

Then, Charlie can easily get e (Q_A, d_B) from the following equality.

$$e(Q_A, d_B) = \left(\frac{\sigma}{t}\right)^v .$$

Using the element e (Q_A, d_B), Charlie is able to impersonate Alice (or Bob) to generate a designated verifier signature on any message by performing the following:

- **Sign**: To sign a message m' for Bob on behalf of Alice (or Bob), Charlie performs as below.

1. Choose a random value $r' \in Z_q*$ and compute $t' = e(P, Q_B)^{k'}$.

2. Set $h' = H_2(m', t')$.

3. Compute $\sigma' = t'e(Q_A, d_B)^{h'}$.

The forged designated verifier signature on the message m' is (t', σ').

- **Verify:** The forged message-signature pair (m' , (t', σ')) can be accepted by Bob since the verifying equality $\sigma' = t'e(Q_A, d_B)^{h'}$ always holds.

Hence, Kang et al.'s scheme is not secure against a universal forgery. That is, in their scheme, an adversary can forge a signature on any message after having a designated verifier signature.

3.2 Violation of Strongness Property

Kang et al. claimed that their scheme is a strong DVS scheme and no third party can verify the validity of a signature. However, we show their scheme can not satisfy this property.

Assume (t, σ) is a signature on a message m. A third party who intercepts (t, σ) can get $e\ (Q_A, d_B)$ by using the above attack technique. With the value $e\ (Q_A, d_B)$, the third party runs the IBSDVS-Verify algorithm and can easily verify the validity of the subsequent signatures without the secret key of the designated verifier. This violates the property of strong designated verifier signatures.

3.3 Delegatability Attack

Kang et al.'s scheme is insecure against delegatability attack, since the signer Alice can delegate her signing capability to any third party without disclosing her secret key. To do so, she only sends a value $e(Q_B, d_A)$ to a third party. Using $e(Q_B, d_A)$, the third party can easily generate a designated verifier signature on any message m as follows:

1. Choose a random value $k \in Z_q{}^*$ and compute $t=e(P, Q_B)^{k}$.
2. Set $h=H_2\,(m, t)$.
3. Compute $\sigma=te(Q_B, d_A)^h$.

The signature on the message m is (t, σ) and it is able to pass the signature verification.

4 A Secure IBSDVS Scheme

In this section, we provide a modification for Kang et al.'s scheme. Unlike their scheme, ours achieves all security requirements of strong designated verifier signatures and it satisfies the properties of unforgeability, non-transferability privacy, source hiding, and non-delegatability.

Without losing generality, we only describe the IBSDVS-Sign, IBSDVS-Verify and IBSDVS-Simulation algorithms, and other algorithms are the same as those defined in [9].

- **IBSDVS- Sign**: To sign a message *m* for Bob, Alice performs as below.

1. Choose a random value $k \in Z_q{*}$ and compute t=kQ_A.
2. Set h=H_2 (m, t).
3. Compute T= (k + h)d_A and σ =e(T, Q_B).

The signature on the message *m* is (t, σ).

- **IBSDVS-Verify**: Given *params*, Alice's public key Q_A and the signature (t, σ) on *m*, Bob performs as follows:

1. Set h=H_2 (m, t).
2. Check whether $\sigma \overset{?}{=} e(t + hQ_A, d_B)$ holds with equality. If so, then accept the signature. Otherwise, reject it.

- **IBSDVS-Simulation**: Bob can produce the signature (t, σ) intended for himself, by performing the following:

1. Choose a random value $k' \in Z_q{*}$ and compute $t' = k'Q_A$.
2. Set $h' = H_2(m,t')$.
3. Compute $\sigma' = e(t' + h'Q_A, d_B)$.

Then, the tuple (t',σ') is a valid signature on message *m*.

5 Security Analysis

1) Correctness

$$\begin{aligned}\sigma &= e(T, Q_B)\\ &= e\big((k+h)d_A, Q_B\big)\\ &= e\big((k+h)Q_A, d_B\big)\\ &= e\big(kQ_A + hQ_A, d_B\big)\\ &= e(t + hQ_A, d_B)\end{aligned}$$

2) ***Strongness***

The IBSDVS-Verify algorithm of our scheme requires the designated verifier Bob's private key d_B, and no one but Bob can perform the signature verification even if the value $e(Q_A, d_B)$ is disclosed. Thus, our scheme is a strong IBSDVS scheme.

3) Unforgeability

To prove the property of unforgeabilty, we review a computational hard problem related to our scheme.

Computational Bilinear Diffie-Hellman Problem (CBDHP) in groups (G_1, G_2) is defined as follows: given (P, aP, bP, cP) for some unknown values $a, b, c \in Z_q{}^*$, to compute $v \in G_2$ such that $v=e(P, P)^{abc}$.

Theorem 1. Our IBSDVS scheme is unforgeable if the assumption of the CBDH in G_1 is intractable. That is, if a valid IBSDV signature can be generated without the knowledge of the private keys of Alice and Bob, there exists an algorithm $\mathcal{C}$ that can solve the CBDH problem in a polynomial time.

Proof. If there is an adversary $\mathcal{F}$, which is able to forge a designated verifier signature, we can build an algorithm $\mathcal{C}$, which can solve the CBDH problem with non-negligible probability. Algorithm $\mathcal{C}$ takes as inputs P, aP, bP, $cP \in \mathrm{G}_1$, where $a, b, c \in Z_q{}^*$ are unknown to $\mathcal{C}$, and $\mathcal{C}$ attempts to extract $e(P, P)^{abc}$ after interacting with $\mathcal{F}$. In our setting, $\mathcal{C}$ sets $Q_A=aP$, $Q_B=bP$ and $P_0=cP$, and gives *params* $\{G_1, G_2, P_0, H_1, H_2\}$ to $\mathcal{F}$.

For simplicity, we assume that H_1 queries are distinct and any query involving an identity ID comes after a H_1 query on the identity ID.

- **H_1-queries** on an input ID_i: C recovers a list H_1^{list} and returns the previously defined value if it exists. Otherwise, C acts as follows:

$$Q_i = H_1(ID_i) = \begin{cases} aP, & if \;\; ID_i = ID_A \\ bP, & if \;\; ID_i = ID_B \\ t_iP, & otherwise, \; t_i \in Z_q^* \end{cases}$$

Then, $\mathcal{C}$ adds (ID_i, Q_i, t_i) into the list H_1^{list} and returns the value Q_i to $\mathcal{F}$.

- **Key-Extract queries** on an input ID_i: $\mathcal{C}$ recovers the corresponding tuple (ID_i, Q_i, t_i) from the list H_1^{list} and performs as below:

1) If $ID_i \neq ID_A$ or ID_B, then returns $d_i = t_icP$ to $\mathcal{F}$ and inserts (ID_i, Q_i, d_i) to a list E^{list}.

2) Otherwise, $\mathcal{C}$ aborts and outputs "failure".

- H_2-**queries** on input (m_i , t_i): $\mathcal{C}$ picks a random $h_i \in Z_q$* and returns it to $\mathcal{F}$, and adds the tuple (m_i , t_i, h_i) to a list H_2^{list}.

- **IBSDVS-Sign queries** on a message m and a signer/designated verifier's identity ID_i / ID_j, $\mathcal{C}$ acts as below:

  - If $ID_i \neq ID_A$ or ID_B, $\mathcal{C}$ recovers the corresponding tuple (ID_i, Q_i, d_i) from the list E^{list} and computes as follows:

    1. Choose a random value $k \in Z_q$* and compute $t=kQ_i$.
    2. Set $h=H_2(m, t)$.
    3. Compute $T= (k+h)d_i$ and $\sigma = e(T, Q_j)$.

  Then, returns the designated verifier signature (t, σ) to $\mathcal{F}$.

- If $ID_j \neq ID_A$ or ID_B, $\mathcal{C}$ recovers the corresponding tuple (ID_j, Q_j, d_j) from the list E^{list} and computes as follows:

1. Choose a random value $k \in Z_q{}^*$ and compute $t=kQ_i$.
2. Set $h=H_2(m, t)$.
3. Compute $\sigma= e(t+hQ_i, d_j)$.

Then, returns the signature (t, σ) to $\mathcal{F}$.

- Otherwise, $\mathcal{C}$ stops the simulation.

- **IBSDVS-Verify queries** on input a signature (t, σ) on a message m and a signer/designated verifier's identity ID_i / ID_j. C checks whether $\{ID_i, ID_j\} = \{ID_A, ID_B\}$ holds. If it holds, quits it. Otherwise, $\mathcal{C}$ finds the designated verifier's private key d_j and verifies the validity of the signature by the IBSDVS-Verify algorithm.

Finally, $\mathcal{F}$ outputs a valid IBSDV signature (t^*, h^*, σ^*) on a message m^* with the signer's identity $ID_i{}^*$ and the designated verifier's identity $ID_j{}^*$. If $\{ID_i{}^*, ID_j{}^*\} \neq \{ID_A, ID_B\}$, $\mathcal{C}$ outputs "failure" and aborts. Otherwise, $\mathcal{C}$ finds the tuple (m^*, t^*, h^*) in $H_2{}^{list}$ and replays $\mathcal{F}$ with the same random tape but different choices of H_2, as done in the forking lemma [10]. $\mathcal{C}$ gets another forgery (m^*, (t^*, h', σ')) such that $h' \neq h^*$. Afterwards, since

$$\sigma^* = e(t^* + h^* Q_A, d_B)$$

$$\sigma' = e(t^* + h' Q_A, d_B),$$

$\mathcal{C}$ has a value $\left(\dfrac{\sigma^*}{\sigma'}\right) = e\big((h^* - h')Q_A, d_B\big)$.

Assume that $h = (h^* - h')^{-1} (\mathrm{mod}\, q)$. Then $\mathcal{C}$ outputs $\left(\dfrac{\sigma^*}{\sigma'}\right)^h$ as the solution to the CBDH problem because $\left(\frac{\sigma *}{\sigma'}\right)^h = e(Q_A, d_B) = e(aP, bcP) = e(P,P)^{abc}$. □

4) Non-transferability

Our scheme achieves the property of non-transferability privacy because the designated verifier Bob can always simulate the received signature (*t, σ*) by generating a valid signature. More precisely, he can compute $t' = k'Q_A$ with a random value k' $\in Z_q*$ and compute $\sigma' =$ $e(t' + H_2(m,t')Q_A, d_B)$ for a message *m* such that (t', σ') passes the signature verification. Obviously, the distribution of (t', σ') is perfectly indistinguishable from that of an original signature generated by algorithm IBSDVS-Sign. As a result, Bob cannot prove to a third party that the signature (*t, σ*) was produced by Alice.

5) Source hiding

Even if Alice's private key d_A and Bob's private key d_B are known to a third party, the third party cannot identify whether d_A or d_B has been used in the construction of the term *σ*, as $\sigma = e((k+h)d_A, Q_B) = e((k+h)Q_A, d_B)$.

6) ***Non-delegatability***

Our scheme satisfies the property of non-delegatability. In fact, even if Alice gives the value $e(Q_B, d_A)$ or the other derivative of her private key to a third party, the third party can not generate a valid designated verifier signature (*t, σ*) because the construction of the term *σ* requires Alice's private key.

6 Conclusion

We first show that Kang e t al.'s IBSDVS scheme [9] is not secure against both universal forgery attack and delegatability attack. Moreover, we find that their IBSDVS scheme can not satisfy strongness property. And then we present an improved scheme for scheme [9] and the improved scheme satisfies properties of strongness, unforgeability, non-transferability, source hiding, and non-delegatability.

References

[1] M. Jakobsson, K. Sako, K. R. Impaliazzo. Designated verifier proofs and their applications. In Eurocrypt 1996, LNCS 1070, Springer-Verlag, 1996, pp. 143-154.

[2] S. Saeednia, S. Kremer, O. Markovitch. An efficient strong designated verifier signature scheme. CICS 2003, LNCS 2971, Springer-Verlag, 2003, pp. 40-54.

[3] F. Laguillaumie, D. Vergnaud. Designated verifier signatures: anonymity and efficient construction from any bilinear map, in: SCN2004, LNCS, vol. 3352, Springer-Verlag, 2004, pp. 105–119.

[4] F. Laguillaumie, D. Vergnaud. Multi-designated verifiers signatures, in: ICICS 2004, LNCS 3269, Springer-Verlag, 2004, pp. 495–507.

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

[6] K. Kumar, G. Shailaja, A. Saxena. Identity based strong designated verifier signature scheme. http//www.eprint.iacr.org/2006/134.

[7] J. Zhang, J. Mao. A novel ID-based designated verifier signature scheme, Information Science, 178(3), 2008, pp. 766-773.

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

[9] B. Kang, C. Boyd, E. Dawson. A novel identity-based strong designated verifier signature scheme, The Journal of Systems and Software (2008), doi: 10.1016/j.jss.2008.06.014.

[10] M. Bellare, G. Neven. Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma. ACM-CCS 2006, 390-399.