TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 10, October 2014, pp. 7343 ~ 7352
DOI: 10.11591/telkomnika.v12i8.5202

Provably Secure and Efficient ID-based Strong Designated Verifier Signature Scheme with Message Recovery

Min Li
Sichuan Normal University of China, Chengdu, Sichuan
University of Electronic Science and Technology of China
email: lm_turnip@126.com

Abstract

Many ID-based strong designated verifier signature schemes have been proposed in recent years. However, most of them did not give the rigorous security proofs and did not satisfy the strongness property that anyone except the designated verifier cannot check the validity of a designated verifier signature, In addition, considering some special applications, these schemes have larger data size of communication. To overcome those problems, exploiting message recovery techniques which are regarded as a useful method to shorten ID-based signatures' size, we put forward an efficient ID-based strong designated verifier signature schemes with message recovery and give its rigorous security proof in the random oracle model based on the hardness assumptions of the computational Bilinear Diffie-Hellman problem in this paper. To the best of our knowledge, it is the first ID-based strong designated verifier signature schemes with message recovery and rigorous security proofs. Due to its merits, it can be used in some special environments where the bandwidth is one of the main concerns, such as PDAs, cell phones, RFID etc.

Keywords: *identity-based public key cryptography, designated verifier signature, message recovery, bilinear pairings, random oracle model*

1. Introduction

Digital signature as one important primitive in cryptography, which can provide data integrity, authentication and non-repudiation, has many practical applications in the real world, such as electronic commerce, electronic government etc. However, in some special environments, signatures with special properties are always desirable. For example, in some scenarios such as E-voting, call for tenders and software licensing, the public verification of an ordinary signature is not desired, since the signer may not want to the recipient of a digital signature to transfer the conviction to a third party at will.

To address this problem above, Chaum and Van Antwerpen introduced undeniable signatures [2, 3] which allowed a signer to completely control his signatures. In undeniable signatures, the verifier (Bob) can not check the validity of the signature given by the signer (Alice) by himself. Instead, Alice participates in the scheme to prove the validity (or invalidity) of the signature to Bob by means of an interactive protocol. Nevertheless, Alice can only decide when to prove, but not who to verify. Hence, the conviction can be transferred to anyone else. Motivated by the above problem, Jakobsson et al. [4] introduced the concept of designated verifier signature (DVS) scheme in Eurocrypt 1996. A DVS scheme makes it possible for a signer Alice to convince a designated verifier Bob that Alice has signed a message in such a way that Bob can not transfer the conviction to a third party Cindy. This is called non-transferability, and is usually achieved by enabling Bob the capability of efficiently simulating a signature which is indistinguishable from Alice's.

In order to enhance the signer's privacy, Jakobsson et al. also introduced a stronger version of DVS in the same work [4]. It is usually called strong designated verifier signature (SDVS) scheme, in which no third party can even check the validity of a designated verifier signature, since the verification of the signature requires the designated verifier's private key.

Received December 6, 2013; Revised July 18, 2014; Accepted August 2, 2014

Since the notion of SDVS proposed by Jakobsson et al. in [4], many SDVS schemes have been put forward in the literature. In 2003, Saeednia et al. [5] firstly formalized the notion of SDVS and proposed an efficient scheme in the same paper. In 2004, Susilo et al. [6] proposed the first strong designated verifier signature scheme in identity-based public key cryptosystem that was first introduced by Shamir [1] in 1984 to solve the problems of certificate management in public key infrastructure (PKI). Due to its advantage in contrast to PKI, several new ID-based SDVS (IBSDVS) have been proposed in the new setting recently. In 2008, Zhang et al. [8] proposed a novel IBSDVS scheme by combining ID-based public key cryptosystem with the designated verifier signature. In their work, they claimed that their scheme was a strong designated verifier signature, that is to say, no third party can check the validity of a designated verifier signature generated by the signer. In 2009, however, Kang et al. [9] found that Zhang et al.'s scheme can not satisfy the strongness property as they claimed in [8]. In the same paper [9], they presented a new IBSDVS scheme and ID-based designated verifier proxy signature scheme (IBDVPS) based on the new IBSDVS scheme. In the meanwhile, they also put forward a novel IBSDVS scheme [10] with security proofs in the random oracle model based on Bilinear Diffie-Hellman assumption. Unfortunately, Lee et al. [11] showed that Kang et al.'s new schemes in [9] are universally forgeable in 2010, that is, anyone can generate a signature on an arbitrarily chosen message without the secret key of either the signer or the designated verifier. To overcome these flaws, they also presented a new IBSDVS and IBDVPS scheme and give the formal security proofs in the random oracle model [14, 15] in the same paper. In 2008, Huang et al. [7] also proposed an efficient IBSDVS scheme which is secure based on a stronger assumption, i.e. Gap Bilinear Diffie-Hellman, the size of the signature in their scheme is very short compared to all the existing schemes. However, the signature in their scheme is short of randomicity because the signatures on the same message are always identical in every time signature generation procedure.

As far as we know, all these existing schemes except the schemes in [7, 11, 16] can not support the strongness property of the SDVS, and the signing messages in all these schemes always need to be transmitted together with the signatures. Thus, these schemes have a large total communication cost, for which they maybe can not efficiently used in some special environments where low- communication and low-computation cost are usually required.

To solve those above problems, combining the message recovery techniques presented in [12], we firstly put forward an efficient IBSDVS scheme with message recovery (IBSDVSMR), and prove its security in the random oracle model based on Bilinear Diffie-Hellman assumption. In the prosed scheme, the message is embedded in a signature and can be recovered from the received signature. Hence, it results in more bandwidth saving. Obviously, it has the advantage of small data size of communication.

Due to its merits, the proposed scheme in this paper is not only quite efficient with respect to computation cost, but also very small with respect to the total length of the signing message and the appended signature, i.e. communication cost. For these advantages, the proposed scheme with message recovery is very useful for the environments where bandwidth is one of the main concerns. For instance, on wireless devices (e.g. PDAs, cell phones, RFID chips and sensors) where battery life is the main limitation, communicating even one bit of data usually uses significantly more power than executing one 32-bit instruction [13]. Reducing the number of communication bits saves power and is important for increasing the battery life.

The rest of this paper is organized as follows. In section 2, we first give some preliminaries, including bilinear maps and some related hard problems, the Model of ID-based strong designated verifier signature scheme with message recovery and so on, and then we put forward an efficient concrete scheme and give its security proof in the random oracle model in Section 3 and section 4, respectively; In Section 5, we compare our schemes with some existing schemes presented in [8, 10, 11]; Finally, we end this paper with a brief conclusion.

2. Research Method

In this section, we briefly review some fundamental backgrounds required through this paper, including bilinear maps, hardness assumptions, the notion of IBSDVSMR and its security definitions etc.

2.1. Some Notations

For convenience, we list some notations with their meanings throughout this paper in Table 1.

Table 1. Notations and their Meanings

Notation	Meaning
$a \,\|\|\, b$	a concatenation of two strings a and b
$\oplus$	X-OR computation in the binary system
$[x]_2$	the binary notation of $x \in Z$
$[y]_{10}$	the decimal notation of $y \in \{0,1\}^*$
$\|\alpha\|_{l_1}$	the first l_1 bits of α from the right side
$_{l_2}\|\alpha\|$	the first l_2 bits of α from the left side

2.2. Bilinear Pairings

Let $(G_1,+)$ be a cyclic additive group of prime order q and $(G_2,\cdot)$ be a cyclic multiplicative group of the same order q, $|q|=l_1+l_2$. We assume that the discrete logarithm problems in both $(G_1,+)$ and $(G_2,\cdot)$ are intractable. Let $e: G_1 \times G_1 \to G_2$ be a bilinear map with the following properties:

(1) Bilinear: $e(aP,bQ)=e(P,Q)^{ab}$, for all $P,Q \in G_1$, and $a,b \in Z_q^*$.

(2) Non-degenerate: There exists $P \in G_1$ such that $e(P,P) \neq 1$.

(3) Computable: There exists an efficient algorithm to computer $e(P,Q)$ for any $P,Q \in G_1$.

A bilinear map satisfying the properties above is named an admissible bilinear map. It can be obtained from the modified Weil and Tate pairings. Followed by are the hardness assumptions used in the security proofs:

Definition 1. Bilinear Diffie-Hellman Problem: The BDH problem is to computer $e(P,P)^{abc}$ when given (P,aP,bP,cP) for some unknown $a,b,c \in Z_q^*$.

Definition 2. Bilinear Diffie-Hellman assumption: Suppose that G is a BDH parameter generator, *Adv*$_G$(B) is the advantage that an algorithm B has in solving the BDH problem. *Adv*$_G$(B) is defined to be the probability that the algorithm B outputs $e(P,P)^{abc}$ when the inputs to algorithm are G_1, G_2, P, aP, bP, cP, where (G_1, G_2, e) is the output of G for sufficiently large security parameter *k*, P is a random generator of G_1, and a,b,c are randomly chosen from Z_q^*. The BDH assumption is that *Adv*$_G$ (B) is negligible for all efficient algorithms B.

2.3. Definition of IBSDVSMR

An IBSDVS scheme with message recovery is a tuple of five polynomial time algorithms as follows:

Setup: This algorithm takes a security parameter *k* as input and returns the system parameters *Params* and a secret master key *master-key*.

Key Extraction: This algorithm takes system parameters *Params*, *master-key* and a user's identity ID_i as input, and then returns the private key Sk_i with respect to the identity ID_i.

Signature Generation: This algorithm takes the system parameters *Params*, a message *m*, a signer's identity ID_A, his corresponding private key Sk_A and the designated verifier's public key ID_B as inputs, and then it outputs a valid signature σ on message *m*.

Signature Verification: This algorithm takes the system parameters *Params*, signature σ, the signer's identity ID_A, the designated verifier's identity ID_B and private key Sk_B as inputs. It outputs 1 if the signature is valid, the signing message can be recovered successfully in this case. Otherwise outputs 0.

Transcript simulation: The designated verifier runs this algorithm to produce identically distributed transcripts which are indistinguishable from the signature generated by the signer.

2.4. Security properties of IBSDVSMR

As defined in [10, 11], the IBSDVS scheme with message recovery should satisfy some main security properties, which are described as follows:

a) **Correctness:** A properly produced IBSDVSMR must be accepted by the signature verification algorithm.

b) **Non-Transferability:** The non-transferability means that any designated verifier can not transfer the conviction to any third party, that is, the designated verifier can not prove to a third party that the signature was produced by the signer or by himself. This is accomplished by a transcript simulation algorithm through which the designated verifier can produce an indistinguishable signature from the one generated by the real signer.

c) **Strongness:** Given a signature, the verification procedure requires the secret key of the designated verifier, that is, any third party can not check the validity of the signature.

d) **Source hiding:** Given a signature on message *m*, it is infeasible to tell apart the signature is produced by the original signer or the designated verifier on earth even if one knows both the secret keys.

e) **Unforgeability:** It is computationally infeasible to construct a valid IBSDVSMR without the knowledge of the private key of either the signer or the designated verifier. The formal definition of existential unforgeability of IBSDVSMR is modeled by the following game between an adversary A and a challenger C.

Game: The game is executed between a challenger C and an adaptively chosen-message and chosen-identity adversary A.

Setup: The challenger C runs the Setup algorithm to generate the system parameters Params and the system master key *master-key*. Then he sends *Params* to adversary A while keeps *master-key* secret.

Queries: A adaptively issues the following queries in a polynomial bounded number of times.

1) **Key extraction queries:** When receiving private key query on identity ID_i, C runs the Key Extraction algorithm and returns the private key Sk_i .

2) **Sign queries:** On receiving a signature query on message *m* for a signer ID_i and a designated verifier ID_j, C runs the Signature Generation algorithm and returns a valid signature σ on message *m*.

3) **Verify queries:** On receiving a verify query on signature σ for the signer ID_i and the designated verifier ID_j, C runs the Signature Verification algorithm and outputs 1 if the signature is valid. In this case, C recovers the message from the signature and returns it. Otherwise, outputs 0.

4) **Forgery:** Eventually, A outputs a tuple (m^*, σ^*, h^*) with the signer ID_i^* and the designated verifier ID_j^* . We say that A wins the game if the follow conditions are all satisfied:

(1) σ^* is a valid signature on messages m^* with the signer ID_i^* and the designated verifier ID_j^* .

(2) During the simulation, ID_i^* and ID_j^* have never been submitted to the **Key extraction queries**.

(3) m^* has never been queried during the **Sign queries** with the signer ID_i^* and the designated verifier ID_j^* .

Definition 3. An IBSDVSMR is existentially unforgeable against adaptively chosen-message and chosen-identity attacks if the success probability of any polynomial bounded adversary A in the above game is negligible.

3. Research Method

3.1. The Efficient IBSDVSMR Schemes

In this part, we present the first efficient IBSDVS schemes with message recovery, which is not only much efficient but also can support the strongness property. In this new signature scheme, it can deal with messages of some fixed length (ie., $m \in \{0,1\}^{l_1}$ for some fixed integer l_1), each algorithm of which is specified as follows:

Setup: Given a security parameter *k*, the KGC generates a cyclic additive group $(G_1, +)$ of prime order q, a multiplicative group $(G_2, \square)$ of the same prime order, and an admissible bilinear map $e: G_1 \times G_1 \to G_2$. The KGC also chooses four cryptographic hash functions: $H_1: \{0,1\}^* \to G_1, H_2: \{0,1\}^* \times G_2 \to \{0,1\}^{l_1+l_2}, F_1: \{0,1\}^{l_1} \to \{0,1\}^{l_2}, F_2: \{0,1\}^{l_2} \to \{0,1\}^{l_1}$ a random $s \in Z_q^*$ and a generator P of G_1, and computes $P_{Pub} = sP$, where s / P_{Pub} is the private/public key pair of KGC, and then publishes the system parameters *Params*:

$$\{G_1, G_2, e, q, P, P_{Pub}, H_1, H_2, F_1, F_2\}$$

Key Extraction: Given an identity ID_i, KGC computes the corresponding private key $Sk_i = sH_1(ID_i) = sQ_i$, where $Q_i = sH_1(ID_i)$, and then sends it to the user with identity ID_i via a secure channel. In this scenario, the signer Alice with identity ID_A has the private key $Sk_A = sQ_A$, and the designated verifier Bob has his private key $Sk_B = sQ_B$.

Signature Generation: To sign a message $m \in \{0,1\}^{l_1}$, the signer Alice with private key Sk_A performs as follows (here we define $g = e(Sk_A, Q_B)$ and it can be pre-computed):

(1) Choose a random element $r \in Z_q^*$, and compute $\alpha = H_2(ID_A, ID_B, g^r) \in \{0,1\}^{l_1+l_2}$, where ID_A and $ID_B \in \{0,1\}^*$.

(2) Compute $\beta = F_1(m) \| (F_2(F_1(m)) \oplus m) \in \{0,1\}^{l_1+l_2}$ and $h = [\alpha \oplus \beta]_{10}$.

(3) Compute $V = (r-h)Sk_A$, and output $\sigma = e(V, Q_B)$. The signature on message *m* is (σ, h).

Signature Verification: Given system parameters *Params*, identity ID_A and the signature (σ, h), the designated verifier Bob with private key Sk_B performs as follows:

(1) Compute $\alpha' = H_2(ID_A, ID_B, \sigma \cdot e(Q_A, Sk_B)^h)$.

(2) Compute $\beta' = [h]_2 \oplus \alpha'$.

(3) Recover the message $m' = |\beta'|_{l_1} \oplus F_2({}_{l_2}|\beta'|)$.

(4) Output 1 and accept (σ, h) as a valid signature of message $m'(=m)$ if and only if $F_1(m') =_{l_2} |\beta'|$.

Transcript simulation: The designated verifier Bob can produce the distinguishable signature $\hat{\sigma}$ intended for himself by doing the following steps:

(1) Randomly choose $\hat{r} \in Z_q^*$, then compute $\hat{U} = e(Sk_B, Q_A)^{\hat{r}}$ and $\hat{\alpha} = H_2(ID_A, ID_B, \hat{U})$.

(2) Compute $\hat{\beta} = F_1(m) \| (F_2(F_1(m)) \oplus m)$ and $\hat{h} = [\hat{\alpha} \oplus \hat{\beta}]_{10}$.

(3) Compute $\hat{\sigma} = e(Sk_B, (\hat{r} - \hat{h})Q_A)$. Then $(\hat{\sigma}, \hat{h})$ is also a valid signature on message *m*.

3.2. Securit Analysis

In this section, we will give security analysis of our proposed scheme.

a) **Correctness:** Given a signature pair (σ, h), the correctness of the proposed scheme can be proved as follows:

$$\sigma \cdot e(Q_A, Sk_B)^h$$
$$= e((r-h)Sk_A, Q_B) \cdot e(Q_A, Sk_B)^h$$
$$= e(Sk_A, Q_B)^r \cdot e(Sk_A, Q_B)^{-h} \cdot e(Q_A, Sk_B)^h$$
$$= g^r \cdot e(Sk_B, Q_A)^{-h} \cdot e(Q_A, Sk_B)^h = g^r$$

If (σ, h) is a valid signature, then we have $H_2(ID_A, ID_B, g^r) = \alpha$ and $[h]_2 \oplus \alpha = \beta = F_1(m) || \ (F_2(F_1(m)) \oplus m)$. Thus, we obtain $|\beta|_{l_1} \oplus F_2({}_{l_2}|\beta|) = m$.

Finally, the integrity of *m* is justified by $F_1(m) =_{l_2} |\beta|$.

b) **Non-Transferability:** The non-transferability means that the designated verifier Bob can not prove to any third party that the signature was produced by the signer Alice or himself. In our scheme, the non-transferability is achieved through the simulation algorithm. In particular, suppose (σ', h') is a signature which is randomly chosen from the set of all valid signatures intended to Bob, then the probability $\Pr[(\sigma, h) = (\sigma', h')] = 1/(q-1)$ since (σ', h') is generated from a randomly chosen value $r \in Z_q^*$. Similarly, it is easy to get that the probability $\Pr[(\hat{\sigma}, \hat{h}) = (\sigma', h')] = 1/(q-1)$. This means that transcripts simulated by Bob and the signatures generated by Alice have the identical distribution, so they are indistinguishable from each other.

c) **Strongness:** Since any information about the private keys can not be obtained from the transcript (σ, h) in our scheme and Bob's private key is required in the verification procedure, any third party without the private key will be unable to check the validity of the signature (σ, h). Thus, the strongness property is achieved in our proposed scheme.

d) **Source hiding:** From the signature generation and simulation algorithms, we can say that even if the third party knows both the signer and the designated verifier's private keys, she still can not identify whether the signature is generated by the original signer or the designated verifier on earth. This is due to the fact that:

$$\sigma = e((r-h)Sk_A, Q_B) = e((r-h)Sk_B, Q_A)$$

Where $h = H_2(ID_A, ID_B, e(Sk_A, Q_B)^r) = H_2(ID_A, ID_B, e(Sk_B, Q_A)^r)$.

Next, we will prove that the proposed scheme is existentially unforgeable against adaptively chosen- message and chosen-identity attacks based on the BDH assumption.

Theorem 1. If there exists an adaptively chosen-message and identity adversary A who can ask at most q_{H_1} times H_1 **queries**, q_{H_2} times H_2 **queries**, q_E times **Key extraction queries**, q_S times **Sign queries**, q_V times **Verify queries**, respectively, and break the proposed scheme in polynomial time *t* with success probability $\varepsilon \geq 10(q_S + 1)(q_{H_2} + q_S) / q$, then there exists an algorithm C that can use A to solve the BDH problem with probability $Adv_c^{BDH}(k)$ in time span *t'*, where $Adv_c^{BDH}(k) \geq (1 - \frac{2}{q_{H_1}})^{q_E + q_V} (1 - \frac{2}{q_{H_1}^2 - q_{H_1}})^{q_S} \frac{2}{q_{H_1}(q_{H_1} - 1)} \varepsilon$, $t' \geq 120686 q_{H_2} qt / 10(q_S + 1) \ (q_{H_2} + q_S) + o(q_{H_1} + q_E + 3q_S + q_V)t_* + o(2q_S + q_V)t_P$, t_* is the time to compute a scalar multiplication in G_1, and t_P is the time to compute a pairing operation on (G_1, G_2).

Proof. Let C be a BDH attacker. He receives a random instance (P,aP,bP,cP) of the BDH problem, his goal is to compute $e(P,P)^{abc}$ after interacting with A in the above **Game**. In our setting, H_1 and H_2 are both regarded as random oracles.

Setup: First, C sets $P_{Pub}=cP$, and generates the system parameters *Params* = $\{G_1,G_2,e,q,P,P_{Pub},\ H_1,H_2,F_1,F_2\}$ running the **Setup** algorithm, then he sends *Params* to the adversary A.

Queries: A adaptively issues the queries to the following oracles in a polynomial bounded number of times. These oracles are all simulated by C. To avoid collisions and consistently respond to the queries, C maintains two lists $L_{H_1}=\{ID_i,Q_i,r_i\}$, $L_{H_2}=\{ID_i,ID_j,U,\alpha\}$ which are initially empty.

(1) H_1 **queries:** Receiving an H_1 query on ID_i, C first scans the list L_{H_1} , then returns the same answer in L_{H_1} if the request has been asked before. Otherwise, C selects a random $r_i\in Z_q^*$, and answers the queries as follows:

$$Q_i=\begin{cases} r_i aP, & \text{if } ID_i=ID_A,\\ r_i bP, & \text{if } ID_i=ID_B,\\ r_i P, & \text{otherwise.}\end{cases}$$

Then C adds (ID_i,Q_i,r_i) to L_{H_1} . In all cases, C returns Q_i as the answer.

(2) H_2 **queries:** When receiving an H_2 query on (ID_i,ID_j,U), C first scans L_{H_2} list , if the request has been asked before, the same answer in the list will be returned. Otherwise, C selects $l\in\{0,1\}^{l_1+l_2}$ at random, and sets $\alpha=l$, then adds (ID_i,ID_j,U,α) to H_2 list L_{H_2} and returns the answer α .

(3) **Key extraction queries:** When A issues a key extraction query on ID_i , C first makes an H_1 query on ID_i, recovers the tuple (ID_i,Q_i,r_i) , and answers the query as follows:

$$Sk_i=\begin{cases} r_i P_{Pub}, & \text{if } ID_i\neq ID_A \text{ } or \text{ } ID_B,\\ \perp, & \text{otherwise.}\end{cases}$$

Then C returns the corresponding answer Sk_i (when $ID_i=ID_A$ or ID_B , C aborts and outputs $\perp$).

(4) **Sign queries:** On receiving a signature query on message *m* for a signer ID_i and the designated verifier ID_j , C does the following steps:

(a) If $ID_i\neq ID_A\ or\ ID_B$, then C recovers (ID_i,Q_i,r_i) from the list L_{H_1} , computes the signer's private key $Sk_i=r_iP_{Pub}=r_icP$, and randomly chooses $t\in Z_q^*$ to generate the signature as follows: $U=e(r_icP,Q_j)^t$

$$h=[H_2(ID_i,ID_j,U)\oplus(F_1(m)\,||\,(F_2(F_1(m))\oplus m))]_{10}\ \sigma=e((t-h)r_icP,Q_j)$$

(b) If $ID_j \neq ID_A \ or \ ID_B$, C recovers (ID_j, Q_j, r_j) from the H_1 list and computes the signer's private key $Sk_j = r_j cP$. Then C selects $t \in Z_q^*$ at random and produces the signature as follows: $U = e(r_j cP, Q_i)^t$

$$h = [H_2(ID_i, ID_j, U) \oplus (F_1(m) \| (F_2(F_1(m)) \oplus m))]_{10} \ \sigma = e((t-h) r_j cP, Q_i)$$

(c) Otherwise, quits the protocol.
Eventually, C returns (σ, h) as the signature on message *m* with the signer's identity ID_i and the designated verifier's identity ID_j .

(5) **Verify queries:** When receiving a verify query on the signature (σ, h) for the signer ID_i and the designated verifier ID_j , C checks weather $(ID_i, ID_j) = (ID_A, ID_B)$ or $(ID_i, ID_j) = (ID_B, ID_A)$ holds. If it holds, then quits it. Otherwise, C recovers (ID_j, Q_j, r_j) from the list L_{H_1} and computes the designated verifier's private key $Sk_j = r_j cP$ to verify the given signature (σ, h) by the algorithm **Signature Verification**. If it is true, C recovers the message, then returns it and outputs 1; or else, C outputs 0.

Forgery: Finally, A outputs a tuple (σ^*, h^*) as the forged signature on message m^* for the signer ID_i^* and the designated verifier ID_j^* with non- negligible probability ε . If $(ID_i^*, ID_j^*) = (ID_A, ID_B)$ or $(ID_i^*, ID_j^*) = (ID_B, ID_A)$, then C outputs (σ^*, h^*) and proceeds. Otherwise, C outputs “fail” and aborts it. Additionally, it is required that the signature (σ^*, h^*) is valid, and that in the simulation ID_i^* , ID_j^* have never been submitted to the **Key extraction queries**, moreover, m^* has never been queried during the Sign queries with the signer's identity ID_i^* and the designated verifier's identity ID_j^* .

If (σ^*, h^*) satisfies all the conditions above, C recovers the tuple $(ID_i^*, ID_j^*, U^*, \alpha^*)$ from the H_2 list, and then replays A with the same random tape but different choice of the hash function H_2 by exploiting the “forking” technique formalized in [15]. On the same message m^* , C gets another valid signature (σ', h') such that $h' \neq h^*$ and $\sigma' \neq \sigma^*$. Then, the BDH problem with instance (P, aP, bP, cP) can be easily solved by C as follows:

(1) If $(ID_i^*, ID_j^*) = (ID_A, ID_B)$, we have $\sigma^* \cdot e(Sk_j^*, Q_i^*)^{h^*} = \sigma' \cdot e(Sk_j^*, Q_i^*)^{h'}$

That is, $\left(\dfrac{\sigma^*}{\sigma'}\right)^{(h' - h^*)^{-1}} = e(Sk_j^*, Q_i^*) = e(r_j^* bcP, r_i^* aP)$

Then, it is easy to get that $\left(\dfrac{\sigma^*}{\sigma'}\right)^{[r_i^* r_j^* (h' - h^*)]^{-1}} = e(P, P)^{abc}$.

(2) If $(ID_i^*, ID_j^*) = (ID_B, ID_A)$, similarly, we obtain

$$\left(\frac{\sigma^*}{\sigma'}\right)^{(h' - h^*)^{-1}} = e(Sk_j^*, Q_i^*) = e(r_j^* acP, r_i^* bP)$$

thus, $e(P, P)^{abc} = \left(\dfrac{\sigma^*}{\sigma'}\right)^{[r_i^* r_j^* (h' - h^*)]^{-1}}$.

Following this, we will compute the probability that C solves the given instance of the BDH problem. C succeeds if:

(1) E_1 : During the simulation, C does not abort any query;

(2) E_2 : A outputs a valid and nontrivial forgery (σ^*, h^*) on message m^* ;

(3) E_3 : E_2 occurs and C does not quit in the forgery pahse.

The probability that C succeeds to solve the BDH problem is $\Pr[E_1 \wedge E_2 \wedge E_3]$, that is, these events happen simultaneously.

From the above game, it is easy to get:

$$Adv_c^{BDH}(k) \geq (1 - \frac{2}{q_{H_1}})^{q_E + q_V} (1 - \frac{2}{q_{H_1}^2 - q_{H_1}})^{q_S} \frac{2}{q_{H_1}(q_{H_1} - 1)} \varepsilon \cdot$$

4. Results and Analysis

In this section, we compare our scheme with those presented in [8], [10-11] from the aspects of the total communication cost, ie. $|signature| + |message|$, and the computation cost required by the signature generation and verification procedures, respectively. In table 2, we denote by P a computation of the pairing operation, S a scalar multiplication in G_1, E an exponentiation in G_2 , and H an unefficient "MaptoPoint" hash function. We also denote the total signature length and the bit length of a point in G_1 by $TS\!-\!L$ and $|G_1|$ (assume that $|G_1| = |G_2|$), respectively. For some operations such as additions in G_1, XOR in the binary system and the common hash function, they are so efficient that they all can be neglected in the comparison.

Table 2. The Comparison of Several Existing Schemes

Schemes	Sign	Verify	*TS-L*	Strongness
Scheme in [8]	4*S*+*H*	3*P*+*H*	3|*G₁*|+*m*	×
Scheme in [10]	2*S*+2*P*+*E*	*P*+*E*	2|*G₁*|+*m*	×
Scheme in [11]	2*S*+2*P*+*E*	2*P*+*S*	2|*G₁*|+*m*	√
Our scheme	*S*+2*P*+*E*	*P*+*E*	|*G₁*|+|*q*|	√

From the comparison above, we can see that, when signing the same messages, our proposed scheme is much more efficient on the whole, and the total communication cost is much less than the others since no message or just partial message needs to be transmitted along with the signature. What is more, our scheme satisfies the strongness property, while others except [11] can not. As far as we know, it is the first IBSDVS scheme with message recovery in the literature, which not only requires low computation power, but also has small communication cost.

5. Conclusion

In this paper, we put forward the first efficient IBSDVSMR scheme and give its security proof in the random oracle model based on the BDH assumptions. The proposed scheme has both low computation and communication cost, and can support the strongness property of SDVS.

References

[1] A Shamir. Identity-based cryptosystems and signature schemes. G.R. Blakely , D. Chaum *Editors*. Crypto 1984, LNCS 196, Springer-verlag, California, USA, 1984: 7-53.

[2] D Chaum. Zero-knowledge undeniable signatures. Advances in Cryptology- Eurocrypt'90, LNCS. *Springer-Verlag.* 1990; 473: 458-464.

[3] D Chaum, H van Antwerpen. Undeniable signatures. Advances in Cryptology- Crypto'89, LNCS. *Springer-Verlag*. 1990; 435: 12-216.

[4] M Jakobsson, K Sako, R Impagliazzo. Designated verifier proofs and their applications. Advances in Cryptology-Eurocrypt'96, *LNCS. Springer-Verlag.* 1996; 1070: 143-154.

[5] S Saeednia, S Kramer, O Markovitch. An efficient strong designated verifier signature scheme. ICISC 2003. *Springer-Verlag, Berlin*. 2003: 40-54.

[6] W Susilo, F Zhang, Y Mu. Identity-based strong designated verifier signature schemes. *Lecture Notes in Computer Science (LNCS)*. 2004; 3108: 313-324.

[7] X Huang, W Susilo, Y Mu, F Zhang. Short identity-based strong designated verifier signature schemes. *Lecture Notes in Computer Science (LNCS).* 2006; 3903: 214-225.

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

[9] BB Kang, C Boyd, E Dawson. Identity-based strong designated verifier signature schemes: attacks and new construction. *Computers & Electrical Engineering*. 2009; 35(1): 49-53.

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

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

[12] R Tso, C Gu, T Okamoto, et al. Efficient ID-Based Digital Signatures with Message Recovery. F. Bao et al. *Editors*. CANS 2007, LNCS 4856, Springer-Verlag. 2007: 47-59.

[13] K Barr, K Asanovic. Energy-aware lossless data compression. *Journal ACM Transaction on Computer Systems* (TOCS). 2006; 24(3): 250-291.

[14] M Bellare, P Rogaway. *Random oracles are practical: a paradigm for designing efficient protocols*. Proceeding CCS' 93 of 1st ACM conference on computer and communications security. 1993: 62-73.

[15] D Pointcheval, J Stern. Security proofs for signature schemes. Eurocrypt 1996. LNCS, 1070. *Springer-Verlag.* 1996: 387-398.

[16] E Yoon. An efficient and secure identity-based strong designated verifier signature scheme. *Information Technology and Control*. 2011; 40(4): 323-329.