International Journal of Hybrid Information Technology
Vol.8, No.3 (2015), pp.165-176
http://dx.doi.org/10.14257/ijhit.2015.8.3.16

A New Efficient Identity-based Strong Designated Verifier Signature Scheme with Message Recovery

Min Li

College Of Computer Science, Sichuan Normal University, Chengdu, China
Email: lm_turnip@126.com,

Abstract

Designated verifier signature (DVS) allows the signer to convince a verifier the validity of a statement but prevent the verifier from transferring the conviction. Strong designated verifier signature (SDVS) is a variant of DVS, which could make it possible for a signer to convince only the designated verifier that the signature is made by the signer. Recently, many strong designated verifier signature schemes have been presented in identity based setting. Unfortunately, most of them can not satisfy the strongness and non-delegatability properties, and cannot be rigorously proved secure. Moreover, for some special applications, these schemes suffer from larger data size of communication. In this paper, we present an efficient identity-based strong designated verifier signature scheme with message recovery, and then rigorously analyze its security properties in the random oracle model. As far as we know, it is the first provably secure identity-based strong designated verifier signature scheme with message recovery, which can be used to sign an arbitrary long message, and can satisfy all the security properties including the existential unforgeability, strongness property, non-transferability and non-delegatability, etc.

Keywords: *Identity-based cryptography, Strong designated verifier signature, Message recovery, Random oracle model*

1. Introduction

With development of the internet, it has brought many conveniences for our daily life. Recently, many forms of traditional business transactions have been performed via internet and the development of e-commerce is quite rapid nowadays. In some specific e-commerce or business transactions, the signer is convinced by nobody, except the designated verifier who can realize the accuracy and legality of message. In this case, non-repudiation of conventional digital signature schemes could not satisfy, so in the special environments, signatures with special properties are always desirable.

In 1996, Jakobsson, *et al.*, introduced the concept of designated verifier signature (DVS) scheme [1]. This kind of scheme makes it possible for a signer to convince only the designated verifier that the signature is made by the signer. In this way the designated verifier cannot transfer the conviction to a third party. In case the designated verifier unintentionally reveals message to the third party, the latter could not be very sure about the accuracy of acquired message. This is since in this kind of scheme the designated verifier has the capability of efficiently simulating a signature that is in- distinguishable from the original signature, which is so-called non-transferability.

To enhance the signer's privacy, Jakobsson, *et al.*, also introduced a stronger version of DVS in the same work [1], which is usually called strong designated verifier signature (SDVS) scheme. In this stronger version of DVS, no third party

ISSN: 1738-9968 IJHIT

can even check the validity of a designated verifier signature, since the verification of the signature needs the designated verifier's private key as input.

Due to the attractive properties of the strong designated verifier signature, many concrete SDVS schemes have been presented in recent years. In 2003, Saeednia, *et al.*, firstly formalized the notion of SDVS [2], and simultaneously proposed an efficient construction. In 2004, Susilo, *et al.*, [3] generalized this notion to the identity based setting, and proposed the first identity based strong designated verifier signature scheme.

Identity based cryptography was first introduced by Shamir [4] in 1984, the motivation of which is to solve the problems of certificate management in the traditional public key infrastructure (PKI). Due to its advantage in contrast to PKI, this new cryptosystem has attracted much attention, and many cryptographic primitives have been proposed in this new setting.

Following Susilo, *et al.*,' seminal work in 2004, several new identity-based SDVS (IBSDVS) schemes have been proposed in this setting. In 2008, Zhang, *et al.*, proposed a novel IBSDVS scheme by combining identity-based public key cryptosystem with the designated verifier signature [5]. In their work, they claimed that their scheme was a strong designated verifier signature. In other words, no third party can check the validity of a designated verifier signature generated by the signer. However, Kang, *et al.*, [6] found that Zhang, *et al.*, scheme cannot satisfy the strongness property as they claimed in [5]. In the same work, they presented a new IBSDVS scheme and identity-based designated verifier proxy signature scheme (IBDVPS) based on the new IBSDVS scheme. Meanwhile, they also put forward a novel IBSDVS scheme in [7] and proved its security in the random oracle model based on Bilinear Diffie-Hellman assumption.

Unfortunately, in 2010 Lee, *et al.*, [8] showed that Kang, *et al.*, new schemes in [6] are universally forgeable, 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 [9, 10] in the same paper. In 2008, Huang, *et al.*, [11] also proposed an efficient IBSDVS scheme which is secure based on a stronger assumption, and 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.

With our best knowledge, most of the existing schemes cannot support the strongness property of the SDVS, except the schemes in [8-9, 11], and many existing schemes cannot achieve the non- delegatability property, which is with respect to (an identity-based variant of) the definition proposed by Lipmaa, *et al.*, [12]. Recently, Huang, *et al.*, [13] revisited IBSDVS, proposed a strictly stronger security model, and presented a concrete construction with non-delegatability. However, the scheme is very complicated and the computation cost is expensive, but it can achieve the stronger security.

Among all these existing schemes, the signing messages all need to be transmitted together with the signatures. In this case, these schemes have a relatively large total communication cost, for which they maybe cannot efficiently used in some special environments where low-communication and low- computation cost are usually required.

By exploiting the message recovery mechanism, we firstly put forward an efficient IBSDVS scheme with message recovery (IBSDVSMR), and give its security proof in the random oracle mode. Actually, the message recovery technique introduce in message recovery signature [14] can make it possible to just transmit the signature itself. Usually, the message can be recovered from the received

signature. In this way, more bandwidth can be saved, so it has the advantage of small data size of communication. For these reasons, it is always regarded as a useful method to shorten signature length.

Thus, our proposed scheme is quite efficient in both the computation cost and the communication cost. Hence, it is quite suitable 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 [15]. Reducing the number of communication bits saves power and is important for increasing the battery life, especially for the people who often do business transactions on cell phones, PDAs, in the age of the e-commerce.

The rest is organized as follows. Section 2 gives some preliminaries, including bilinear maps and some related hard problems, the model of identity-based strong designated verifier signature scheme with message recovery, *etc.*, Section 3 and section 4 give an efficient concrete construction and its security proof in the random oracle model, respectively. A performance analysis is given in section 5. At last, the paper is ended with a brief conclusion.

2. Preliminaries

In this section, we briefly recall some fundamental backgrounds used in our work, such as bilinear maps, hardness assumptions, the notion of IB-SDVSMR and the security definitions, *etc.*

Before going ahead, we first give some notations used in our following work, they are listed as follows:

$[x]_2$: The binary notation of $x \in Z$;

$[y]_{10}$: The decimal notation of $y \in \{0,1\}^*$;

$a \| b$: A concatenation of two strings a & b;

$\oplus$: X-OR operation in the binary system;

$|\alpha|_{l_1}$: The first l_1 bits of α from the right side;

$_{l_2}|\beta|$: The first l_2 bits of β from the left side.

2.1 Bilinear Pairings

Let $(G_1,+)$ and $(G_2,\square)$ be a cyclic additive group and a cyclic multiplicative group of prime order q, respectively, where $|q|=l_1+l_2$. Let $e: G_1 \times G_1 \to G_2$ be a bilinear map with the following properties:

1) Bilinearity: $e(aP,bQ)=e(P,Q)^{ab}$, for all $P,Q \in G_1$, and $a,b \in Z_q^*$.
2) Non-degenerality: There exists $P \in G_1$ such that $e(P,P) \neq 1$.
3) Computation feasibility: There exists an efficient algorithm to computer $e(P,Q)$ for any $P,Q \in G_1$.

We call the bilinear map satisfying the properties above admissible, and we can obtain it from the modified Weil or Tate pairing.

In the following, we will give the hardness assumptions used to prove our construction secure.

Definition 1. Bilinear Diffie-Hellman Problem (BDH problem): Given a tuple (P,aP,bP,cP), the BDH problem is to compute $e(P,P)^{abc}$, where $\{a,b,c\}$ are randomly chosen from Z_q^*.

Definition 2. Computational Bilinear Diffie-Hellman assumption (CBDH assumption): Suppose that $\mathcal{G}$ is a BDH parameter generator, $Adv_{\mathcal{G}}(\mathcal{B})$ is the advantage that an algorithm $\mathcal{B}$ has in solving the BDH problem. $Adv_{\mathcal{G}}(\mathcal{B})$ is defined to be the probability that the algorithm $\mathcal{B}$ outputs $e(P,P)^{abc}$ when taking $(G_1, G_2, P, aP, bP, cP)$ as input, where (G_1, G_2, e) is the output of $\mathcal{G}$ 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 CBDH assumption is said that $Adv_{\mathcal{G}}(\mathcal{B})$ is negligible for any efficient algorithms $\mathcal{B}$.

2.2 Definition of IBSDVSMR

An IBSDVS scheme with message recovery consists of five polynomial time algorithms, each of which is described as follows:

Setup: Taking a security parameter *k* as input, it outputs the system parameters *Params* and a secret master key *master-key*.

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

Signature Generation: Taking 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, then it outputs a valid signature σ on message *m*.

Signature Verification: Taking 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 then outputs 1 if the signature is valid, the signing message can be recovered successfully in this case. Otherwise outputs 0.

Transcript Simulation: This algorithm is used by the designated verifier to produce identically distributed transcripts which are indistinguishable from the signature generated by the signer.

2.3 Security Properties of IBSDVSMR

The IBSDVS scheme with message recovery should satisfy the following main security properties:

- **Correctness:** A properly produced IBSDVSMR must be accepted by the signature verification algorithm.
- **Strongness:** Given a signature, the verification procedure requires the secret key of the designated verifier, that is, any third party cannot check the validity of the signature.
- **Non-Transferability:** The non-transferability means that any designated verifier cannot transfer the conviction to any third party, that is, the designated verifier cannot 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 in-distinguishable signature from the one generated by the real signer.
- **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.
- **Non-delegatability:** The non-delegatability property means that if an entity can create a valid signature, it must knows the private key of the signer or the verifier. In other words, any entity cannot generate a valid signature without either the signer's or the designatured verifier's private key.
- **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 $\mathcal{A}$ and a challenger C.

Game: This game played between a challenger C and an adaptively chosen-message and chosen- identity adversary $\mathcal{A}$ mainly consists of the following three phases.

Setup Phase: In this phase, the challenger C runs the Setup algorithm to generate the system parameters *Params* and the system master key *msk*. Then C sends *Params* to the adversary $\mathcal{A}$ while keeps *msk* secret.

Queries Phase: In this phase, the adversary $\mathcal{A}$ can adaptively issues the following three kinds of queries, besides the random oracle queries.

- **Key Extraction Queries:** When the challenger receives a private key query on identity ID_i, C runs the Key Extraction algorithm and returns the private key Sk_i as the response.
- **Sign Queries:** when the adversary asks 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 its output σ as the valid signature on message *m*.
- **Verify Queries:** 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.

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

- σ^* Is a valid signature on messages m^* with the signer ID_i^* and the designated verifier ID_j^*.
- During the simulation, ID_i^* and ID_j^* have never been submitted to the Key extraction queries.
- 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 un- forgeable against the adaptively chosen-message and chosen-identity attacks if the success probability of any probabilistic polynomial time adversary $\mathcal{A}$ in the above game is negligible in the security parameter.

3. The Efficient IBSDVSMR Scheme

In this section, we present the first efficient IBSDVS with message recovery, which can deal with messages of arbitrary lengths; each algorithm is specified as follows:

Setup (1^k): Taking a security parameter 1^k as input, the KGC generates a cyclic additive group $(G_1, +)$ and a multiplicative group $(G_2, \square)$, both of which have the same group order, and also an admissible bilinear map $e: G_1 \times G_1 \to G_2$. In addition, it 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. Then, it sets the the system parameters *Params* and master secret key *msk* as follows:

$Params = \{G_1, G_2, e, q, P, P_{Pub} = sP, H_1, H_2, F_1, F_2\}$ and $msk = s$.

Key Extraction (ID_i, *msk*): Given an identity ID_i, KGC uses *msk* to generate the corresponding private key $Sk_i = sH_1(ID_i) = sQ_i$ for ID_i, where $Q_i = sH_1(ID_i)$, and then

it sends the private key to the user with identity ID_i via a secure channel. In IBSDVSMR, the signer with identity ID_A possesses the private key $Sk_A = sQ_A$, and the designated verifier possesses the private key $Sk_B = sQ_B$.

Signature Generation (m , Sk_A , ID_B): To sign a message $m \in \{0,1\}^*$ for the designated verifier with identity ID_B , the signer with private key Sk_A performs the following steps:

(1) Divide m into $m_1 \| m_2$, where $|m_1| = l_1$.

(2) Compute $\alpha = H_2(m_2, ID_A, ID_B, e(Sk_A, Q_B))$, where ID_A , $ID_B \in \{0,1\}^*$ and $\beta = F_1(m_1) \| (F_2(F_1(m_1)) \oplus m_1)$. Then set $h = [\alpha \oplus \beta]_{10}$.

(3) Randomly choose a random r from Z_q^* , and comput $u = (r+h)Sk_A$ and $v = \frac{h}{r+h} Q_B$. Then output $\sigma = (u, v, h)$ as the signature on message m.

Signature Verification (σ , ID_A , Sk_B): Taking *Params*, partial message m_2 , identity ID_A and the signature (σ, h) as input, the designated verifier possessing private key Sk_B computes as follows:

(1) Verify if the equation $e(u,v) = e(Sk_B, Q_A)^h$ holds.

(2) Evaluate $\alpha' = H_2(m_2, ID_A, ID_B, e(Q_A, Sk_B))$, and compute $\beta' = [h]_2 \oplus \alpha'$.

(3) Recover the message $m_1' = |\beta'|_{l_1} \oplus F_2(_{l_2} | \beta'|)$.

(4) Output 1 and accept σ as a valid signature if and only if the equation above is true and m_1' satisfies $F_1(m_1') =_{l_2} |\beta'|$; otherwise, output 0 and abort.

(5) Recover $m = m_1' \| m_2$.

Transcript Simulation (ID_i, m, Sk_B): The designated verifier cannot prove to a third party that the signature σ was originally generated by the signer. Because he can easily create a distinguishable signature σ' by doing the following steps:

(1) Divide m into $m_1 \| m_2$, where $|m_1| = l_1$.

(2) Compute $\alpha' = H_2(m_2, ID_A, ID_B, e(Q_A, Sk_B))$, where $ID_A, ID_B \in \{0,1\}^*$, and $\beta' = F_1(m_1) \| (F_2(F_1(m_1)) \oplus m_1)$. Then set $h' = [\alpha' \oplus \beta']_{10}$.

(3) Choose $r' \in Z_q^*$, and compute $u' = (r'+h')Sk_B$ and $v' = \frac{h'}{r'+h'} Q_A$. Then $\sigma' = (u', v', h')$ is a valid signature on the message m. It is obvious that σ' satisfies the signature verification algorithm.

4. Security Analyses

In this section, we will give the security analysis of the proposed scheme, including the correctness, strongness, non-transferability, non-delegatability, *etc*.

4.1 Correctness

Given a signature pair $\sigma = (u, v, h)$ and the partial message m_2, the correctness of the proposed scheme is got from the following equations. From the construction, we have:

$$\begin{aligned} e(u,v) &= e((r+h)Sk_A, \frac{h}{r+h} Q_B) \\ &= e(Sk_A, Q_B)^h \\ &= e(Sk_B, Q_A)^h, \end{aligned}$$

and $e(Sk_A, Q_B) = e(Sk_B, Q_A)$.

If $\sigma = (u, v, h)$ is valid, then we have $F_1(m_1) \| (F_2(F_1(m_1)) \oplus m_1) = \beta = [h]_2 \oplus \alpha$, where $\alpha = H_2(m_2, ID_A, ID_B, e(Sk_B, Q_A))$. Thus, we can obtain $|\beta|_{l_1} \oplus F_2(_{l_2}|\beta|) = m_1$.

Finally, the integrity of m_1 is justified by $F_1(m_1) =_{l_2} |\beta|$.

4.2 Strongness

Obviously, the designated verifier's private key is required in the verification procedure, and any information about the private keys cannot be obtained from the transcript $\sigma = (u, v, h)$ in our scheme, so any third party without the private key has no ability to check the validity of the signature σ. Thus, our proposed scheme achieves the strongness property.

4.3 Non-Transferability

The non-transferability in our scheme is achieved via the simulation algorithm. In particular, suppose that $\bar{\sigma} = (\bar{u}, \bar{v}, \bar{h})$ is a signature which is randomly chosen from the set of all valid signatures which is intended to the designated verifier, then the probability $\Pr[(u, v, h) = (\bar{u}, \bar{v}, \bar{h})] = 1/(q-1)$ since $\bar{\sigma} = (\bar{u}, \bar{v}, \bar{h})$ is generated from a random $r \in Z_q^*$. Similarly, it is easy to get that the probability $\Pr[(u', v', h') = (\bar{u}, \bar{v}, \bar{h})] = 1/(q-1)$. This means that the transcripts simulated by the designated verifier and the signatures generated by the real signer have the identical distributions. Therefore, they are indistinguishable from each other for any third party.

4.4 Source Hiding

Through the signature generation and simulation algorithm, it is easy to observe that the third party cannot identify whether the signature is generated by the original signer or the designated verifier on earth, even if he knows both the signer and the designated verifier's private keys. This is because he never know the randomness used in the sign process, and $e(Sk_A, Q_B) = e(Sk_B, Q_A)$ and $e(u, v) = e(Sk_A, Q_B)^h = e(Sk_B, Q_A)^h$.

4.5 Non-delegatability

The non-delegatability means that if an entity can create a valid signature, it must knows the private key of the signer or the verifier. In our scheme, the signature, $u = (r + h)Sk_A$, is generated by using Sk_A, instead of the information like $e(Sk_A, Q_B)$, in contrast to the work [8, 9, 10]. Hence, the third party still cannot generate a valid signature even thouth he can obtain any one-way information about Sk_A and Sk_B, such as $e(Sk_A, Q_B)$ or $e(Sk_B, Q_A)$.

4.6 Unforgeability

In the following, we prove that the proposed scheme is existentially unforgeable against the adaptively chosen-message and chosen-identity attacks based on the hardness assumption defined before.

Theorem 1. Under the CBDH assumption, our proposed IBSDVSMR scheme is existentially unforgeable against the adaptively chosen-message and chosen-identity attacks.

Specifically, if there exists an adaptively chosen-message and identity adversary $\mathcal{A}$ who can break the our scheme in polynomial time t with success probability ε, then we can construct an algorithm C that can use $\mathcal{A}$ as a subroutine to solve the CBDH problem with probability $Adv_c^{CBDH}(k) \geq (1-\frac{2}{q_{H_1}})^{q_E+q_V}(1-\frac{2}{q_{H_1}^2-q_{H_1}})^{q_S}\frac{2}{q_{H_1}(q_{H_1}-1)}\varepsilon$, where q_{H_i} is the maximum number of H_i queries $\mathcal{A}$ can ask, q_E is the maximum number of key extraction queries, q_S is the maximum number of sign queries, and q_V is the maximum number of verify queries,

Proof. Suppose there is an adversary $\mathcal{A}$ that can break our proposed signature scheme, then we can find an algorithm C to solve the CBDH problem. It is constructed as follows. Given a random instance (P, aP, bP, cP) of the CBDH problem, it tries to compute $e(P,P)^{abc}$ through interacting with $\mathcal{A}$ in the above Game. In our setting, H_1 and H_2 are both regarded as random oracles. Without loss of generality, we assume that the H_1 queries are distinct and a H_1 query on the identity ID has been made before any query involving ID.

Setup Phase: C sets $P_{Pub} = cP$, and runs the Setup algorithm to generate the system parameters $Params = \{G_1, G_2, e, q, P, P_{Pub}, H_1, H_2, F_1, F_2\}$. Then, he sends $Params$ to the adversary $\mathcal{A}$.

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

- H_1 **queries:** When receiving an H_1 query on ID_i, C first scans the list L_{H_1}. If the request has been asked before, then C returns the same answer in L_{H_1}; otherwise, C selects a random $r_i \in Z_q^*$, and answers the H_1 query 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 the list L_{H_1}, and returns Q_i as the response.

- **Key Extraction Queries:** When $\mathcal{A}$ asks a key extraction query on ID_i, C first recovers the tuple (ID_i, Q_i, r_i) from the H_1 query list L_{H_1}, and then answers this query as follows:

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

  At last, C returns the corresponding answer Sk_i when $ID_i \neq ID_A$ or ID_B; otherwise, C aborts and outputs $\perp$.

- H_2 **Queries:** When receiving an H_2 query on (m_2, ID_i, ID_j, w), C first scans the L_{H_2} list. If the query on (m_2, ID_i, ID_j, w) has been asked before, the same answer in L_{H_2} will be returned; otherwise, C randomly selects $l \in \{0,1\}^{l_1+l_2}$,

and sets $\alpha = l$. Then, C adds $(m_2, ID_i, ID_j, w, \alpha)$ to the H_2 list L_{H_2} and returns α as the answer.

- **Sign Queries:** When receiving a signature query on message *m* for a signer ID_i and the designated verifier ID_j, C answers this query in the following steps:

(1) If $ID_i \neq ID_A$ *or* ID_B, C recovers (ID_i, Q_i, r_i) from the H_1 query list, computes the signer's private key $Sk_i = r_i P_{Pub} = r_i cP$, and randomly chooses t from Z_q^*. Then, the signature on message *m* is generated as follows:

a) Divides the message *m* into two parts $m_1 \| m_2$, where $|m_1| = l_1$.

b) Computes $w = e(r_i cP, Q_j)$, and then gets $h = [H_2(m_2, ID_i, ID_j, w) \oplus (F_1(m_1) \| (F_2(F_1(m_1)) \oplus m_1))]_{10}$.

c) Evaluates $u = (t+h) r_i cP$ and $v = \frac{h}{t+h} Q_j$.

Then C returns $\sigma = (u, v, h)$ as the signature on message *m*.

(2) 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 creats the signature on *m* as follows:

a) Divides the message *m* into two parts $m_1 \| m_2$, where $|m_1| = l_1$.

b) Computes $w = e(r_j cP, Q_i)$, and then gets $h = [H_2(m_2, ID_i, ID_j, w) \oplus (F_1(m_1) \| (F_2(F_1(m_1)) \oplus m_1))]_{10}$.

c) Evaluates $u = (t+h) r_j cP$ and $v = \frac{h}{t+h} Q_i$.

Then, C returns $\sigma = (u, v, h)$ as the corresponding signature.

(3) Otherwise, C aborts.

- **Verify Queries:** When C receives a verify query on the signature $\sigma = (u, v, h)$ for the signer ID_i and the designated verifier ID_j, C first checks weather $\{ID_i, ID_j\} = \{ID_A, ID_B\}$ holds. If so, then C quits it; otherwise, C first recovers (ID_j, Q_j, r_j) from the list L_{H_1} and then creates the designated verifier's private key $Sk_j = r_j cP$ to verify the given signature $\sigma = (u, v, h)$ by running the Signature Verification algorithm.

Forgery Phase: Eventually, suppose $\mathcal{A}$ outputs a tuple $\sigma^* = (u^*, v^*, h^*)$ as the forged signature on message m^* for the signer ID_i^* and the designated verifier ID_j^* with non-negligible probability ε. If $\{ID_i^*, ID_j^*\} = \{ID_A, ID_B\}$, then C outputs $\sigma^* = (u^*, v^*, h^*)$ and proceeds; otherwise, C outputs "fail" and aborts it. In addition, it is required that the signature $\sigma^* = (u^*, v^*, h^*)$ is valid, and that in the simulation the identities ID_i^* and ID_j^* have never been submitted to the Key extraction queries. Furthermore, the sign query on m^* for the signer's identity ID_i^* and the designated verifier's identity ID_j^* has never been submitted.

If $\sigma^* = (u^*, v^*, h^*)$ satisfies all the conditions above, C recovers the tuple $(m_2, ID_i^*, ID_j^*, w^*, \alpha^*)$ from the H_2 query list, and can solve the CBDH problem by computing $e(u^*, v^*)^{(r_i^* r_j^* h^*)^{-1}}$ or $w^{*(r_i^* r_j^*)^{-1}}$. In particular, it can be obtained as follows:

(1) If $(ID_i^*, ID_j^*) = (ID_A, ID_B)$, we have

$$e(u^*, v^*)^{h^{*-1}} = e(Sk_j^*, Q_i^*) = e(r_j^* bcP, r_i^* aP)$$

Then, it is easy to get that

(2) $e(u^*, v^*)^{(r_i^* r_j^* h^*)^{-1}} = e(P, P)^{abc}$.

If $(ID_i^*, ID_j^*) = (ID_B, ID_A)$, similarly, we can obtain that:

$$e(u^*, v^*)^{h^{*-1}} = e(Sk_j^*, Q_i^*) = e(r_j^* acP, r_i^* bP),$$

Thus, we can obtain:

$$e(P, P)^{abc} = e(u^*, v^*)^{(r_i^* r_j^* h^*)^{-1}}.$$

Note that, we also can obtain $e(P, P)^{abc}$ just from w^* no matter whether $(ID_i^*, ID_j^*) = (ID_A, ID_B)$ or (ID_B, ID_A), this is because $w^* = e(Sk_i^*, Q_j^*) = e(P, P)^{r_i^* r_j^* abc}$.

From the above, it is easy to get that the success probability of $\mathcal{A}$ is

$$Adv_C^{CBDH}(k) \geq (1 - \frac{2}{q_{H_1}})^{q_E + q_V} (1 - \frac{2}{q_{H_1}^2 - q_{H_1}})^{q_S} \frac{2}{q_{H_1}(q_{H_1} - 1)} \varepsilon.$$

5. Performance Analyses

In this section, we will give a performance analysis of our scheme and compare it with some related schemes which cannot realize message recovery.

For convenience, we fist give some symbols used in the following table. We denote a computation of the pairing operation by P, a scalar multiplication in G_1 by S, an exponentiation in G_2 by E, and an inefficient "MaptoPoint" hash function by H. 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.

In our proposed scheme, the sign algorithm does not need any pairing computation and verify algorithm requires only one pairing operation, since the value $e(Sk_A, Q_B)$ or $e(Sk_B, Q_A)$ can be pre-computed and stored ahead of time.

Table 1. Comparison of Several Related Schemes

Schemes	Sign	Verify	TS-L	Strongness	Non-delegatability
Scheme in [5]	4*S*+*H*	3*P*+*H*	3|G_1|+m	×	√
Scheme in [7]	2*S*+2*P*+*E*	*P*+*E*	2|G_1|+*m*	×	×
Scheme in [8]	2*S*+2*P*+*E*	2*P*+*S*	2|G_1|+*m*	√	×
Our scheme	2*S*	*P*+*E*	|G_1|+|*q*|+*m*-l_1	√	√

From the performance analysis in Table 1, 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 just partial message needs to be transmitted along with the signature. Morevoer, our scheme satisfies the strongness property, while others except scheme [8] cannot. However, scheme [8] cannot satisfy the non-delegatability property. To the best of our knowledge, it is the first IBSDVS scheme with message recovery for the messages of arbitrary length, achieving both the strongness property and the non-delegability. Actually, our scheme is the first IBSDVS with message recovery, so we just compare it with some IBSDVS schemes, which do not support the functionality of message recovery.

6. Conclusion

In this paper, we put forward the first efficient IBSDVSMR scheme with message recovery for arbitrary long message, and analyzed its security in the random oracle model. The proposed schemes have both low computation and communication cost, and support not only the strongness property but also the non-delegatability property. In the future, we will try to give some other IBSDVSMR schemes, which have a better performance and a stronger security proved without random oracles.

Acknowledgement

This work was supported by Key Project on the Integration of Industry, Education and Research of Guangdong Province (Grant No. 2012B091000054), and Sichuan Department of Education (Grant No.13ZB0152). This work was also partly supported by the National Nature Science Foundation of China under Grant (No.61373163).

References

[1] M. Jakobsson, K. Sako and R. Impagliazzo, "Designated verifier proofs and their applications. In: Advances in Cryptology-Eurocrypt'96", LNCS, Springer-Verlag, Berlin Heidelberg, vol. 1070, **(1996)**, pp. 143-154.
[2] S. Saeednia, S. Kramer and O. Markovitch, "An efficient strong designated verifier signature scheme", In: ICISC 2003, Springer-Verlag, Berlin, **(2003)**, pp. 40-54.
[3] W. Susilo, F. Zhang and Y. Mu, "Identity-based strong designated verifier signature schemes", ACISP LNCS 3108, Springer-Verlag, Berlin Heidelberg, vol. 3108, **(2004)**, pp. 313-324.
[4] A. Shamir, "Identity-based cryptosystems and signature schemes", In: G.R. Blakely, D. Chaum Editors. Crypto 1984, LNCS 196, Springer-verlag, California USA, **(1984)**, pp.7-53.
[5] J. Zhang and J. Mao, "A novel ID-based designated verifier signature scheme", Information Sciences, vol. 178, no. 3, **(2008)**, pp. 766-773.
[6] B. Kang, C. Boyd and E. Dawson, "Identity-based strong designated verifier signature schemes: attacks and new construction", Computers & Electrical Engineering, vol. 35, no. 1, **(2009),** pp. 49-53.
[7] 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, **(2009)**, pp. 270-273.
[8] J. Lee, J. Chang and D. Lee, "Forgery attacks on Kang, *et al.*, identity-based strong designated verifier signature scheme and its improvement with security proof", Computers & Electrical Engineering, Computer& Electrical Engineering, vol. 36, no. 5, **(2010)**, pp. 948-954.
[9] E. Yoon, "An efficient and secure identity- based strong designated verifier signature scheme", Information Technology and Control, vol. 40, no. 4, **(2011)**, pp. 323-329.
[10] M. Bellare and P. Rogaway, "Random oracles are practical: paradigm for designing efficient protocols", In: First ACM conference on computer and communications security; **(1993)**, New York USA.
[11] X. Huang, W. Susilo, Y. Mu, *et al.*, "Short identity-based strong designated verifier signature schemes. Lecture Notes in Computer Science (LNCS)", Springer-Verlag, Berlin Heidelberg, vol. 3903, **(2006)**, pp. 214-225.
[12] H. Lipmaa, G. Wang and F. Bao, "Designated verifier signature schemes: Attacks new security notions and a new construction", Proceedings of 32th International Colloquium on Automata, Languages and Programming (ICALP), Springer, Berlin Heidelberg **(2005)**, pp. 459-471.
[13] Q. Huang, G. Yang, D. Wong, *et al.*, "Identity-based strong designated verifier signature revisited", The Journal of Systems and Software, vol. 84, no. 1, **(2011),** pp. 120–129.
[14] R. Tso, C. Gu, T. Okamoto, *et al.*, "Efficient ID-Based Digital Signatures with Message Recovery", In: F. Bao, *et al.*, *Editors*. CANS 2007, LNCS 4856, Springer-Verlag, Berlin Heidelberg, **(2007)**, pp. 47-59.
[15] K. Barr and K. Asanovic, "Energy-aware lossless data compression", In: MobiSys 2003. Proceedings of the ACM Conference on Mobile Systems, Applications, and Services, **(2006)**, New York USA.

Authors

Min Li, She received the Ph.D. degree in the University of Electronic Science and Technology of China, in 2014. Her current research interests include Security Protocols, privacy protection, specifically the location privacy in LBS.