A Certificateless Strong Designated Verifier Signature Scheme with Non-delegatability

Yang Chen[1*], Yang Zhao[2*], Hu Xiong[2,3], and Feng Yue[2]
(Corresponding author: Feng Yue)

Sichuan Aerospace Vocational College[1]
No.155, Tiansheng Road, Longquanyi District, Chengdu City, Sichuan Province, China
School of Computer Science and Engineering & University of Electronic Science and Technology of China[2]
No. 4, North Jianshe Road, Chenghua District, chengdu, Sichuan, 610054, China
State Key Laboratory of Information Security, Institute of Software & Chinese Academy of Sciences[3]
No. 19, Yuquan Road, Shijingshan District, Beijing, 100190, China
(Email: yf1513875@163.com)
(Received Nov. 6, 2015; revised and accepted Mar. 6 & Apr. 9, 2016)

Abstract

The designated verifier signature only enables the designated verifier to check the correctness of the signature, while any third party can not verify whether this signature is valid or not. Most of the previous designated verifier signature schemes depend on certificate-based cryptography or identity-based cryptography, while little attention has been paid to the certificateless designated verifier signature scheme which has much more advantages than the previous constructions. In this paper, we propose the first certificateless strong designated verifier signature scheme with non-delegatability. We show that our scheme satisfies the basic properties of a designated verifier signature scheme and resists the two types of adversaries in certificateless cryptography. In addition, the comparison with other existing certificateless SDVS schemes demonstrates the proposed scheme is provided with a good level of security and performance.

Keywords: Certificateless cryptography, designated verifier signature, non-delegatability, strong designated verifier signature

1 Introduction

As we all know, the correctness of the conventional digital signature can be checked by anyone using the signer's public key. However, in some situations such as e-voting [14], e-bidding and software licensing, the signer do not desire the receiver to convince the third party of the signature's authenticity. To settle this problem, Jakobsson et al. [9] proposed the notion of designated verifier signature which can be abbreviated to DVS. The most obvious difference between the conventional digital signature and the DVS is that the designated verifier can not persuade the third party to trust the correctness of the signature in DVS scheme, because the designated verifier is able to construct the signature designated to himself which is indistinguishable from the real signer's signature. Meanwhile, Jakobsson et al. [9] also introduced the conception of the strong designated verifier signature(SDVS) in which the designated verifier's secret key must be used in the verifying phase. Most of existing (S)DVS schemes are based on certificate-based cryptography or identity-based cryptography. Since the public key certificate is involved, the certificate-based (S)DVS schemes bring in massive consumption of certificate management. As to the identity-based (S)DVS schemes, the key escrow problem also causes fatal threats to the users in the scheme. In order to avoid the two inherent flaws mentioned above, the concept of certificateless designated verifier signature scheme is proposed by Huang et al. [6]. Certificateless cryptography is able to avert the utility of public key certificate. Meanwhile, certificateless cryptography ensures the security of user's private key, because the KGC(Key Generation Center) just can get user's partial private key instead of full private key. In this paper, we focus on constructing a certificateless SDVS scheme with non-delegatability.

1.1 Related Works

The notions of DVS and SDVS were firstly proposed by Jakobsson et al. [9] in 1996 and more and more attention was paid to this special signature scheme. In 2003, Saeednia et al. [19] firstly made the formal definition of SDVS and proposed an efficient SDVS scheme without the layer of encryption. In 2004, Susilo et al. [21] introduced the

*These authors contribute equally to this work.

concept of identity-based strong designated verifier signature (IBSDVS) which was built on identity-based cryptography and provided a concrete construction. Because of abandoning the public key certificate, this construction was much more efficient than the certificate-based schemes. As the definition of DVS and SDVS became formalized, some other (S)DVS schemes with new construction methods emerged [7, 10, 18]. Until 2005, Lipmaa et al. [15] figured out a type of attack called delegatability attack on (S)DVS. The general idea of delegatability attack is that the signer and the verifier can illegitimately delegate his ability of signing or verifying to any third party he wants through transferring a common value relating to their private keys to the third party, while the third party disables to extract their private keys from the common value. The proposed delegatability attack makes most of the previous schemes insecure. For the sake of achieving the goal of non-delegatability, [5, 12, 25] were proposed in succession. Unfortunately, Shim et al. [20] figured out the schemes [12, 25] were delegatable and Zhang et al. [26] also proved the scheme [5] was not secure for its delegatability. Recently, Tian et al. proposed a non-delegatable SDVS on elliptic curves [22] and a corresponding identity-based version [23] subsequently. The two schemes were both constructed on the basis of Schnorr digital signature. Until now, they seem to have not been found suffering from the delegatability attack [20].

Appearing later than certificate-based cryptography and identity-based cryptography, certificateless cryptography was firstly proposed by Al-Riyami et al. [1] in 2003. Each user's full private key is constituted by two parts called partial private key and secret value in certificateless cryptography. They are derived from the KGC and the user himself/herself respectively. The user keeps the secret value all the time and the KGC is prohibited from obtaining it. Since the certificateless cryptography was presented relatively late, only serval certificateless (S)DVS schemes were proposed [3, 4, 6, 8, 24]. According to the attack methods in [2], the scheme in [6] suffered from malicious KGC attack. Liu et al. [17] proved the scheme [8] also did not resist malicious KGC attack. Furthermore, utilizing the delegatability attack methods based on [20], we find that the above schemes are subjected to delegatability attack due to the leakage of common value in the signature construction.

1.2 Contributions

In this paper, by means of improving the Schnorr digital signature, we construct the first certificateless strong designated verifier signature scheme with non-delegatability. We formally prove the proposed scheme can resist the two types of attack method including public key replacement attack and malicious KGC attack in certificateless cryptography. The security proofs also contain the properties of non-delegatability and source hiding which are necessary properties of an SDVS. To the best of our knowledge, there is no certificateless SDVS satisfying the property of non-delegatability at present and our scheme is the first one. Besides, We make a comparison with other existing certificateless SDVS schemes to show the proposed scheme possesses a good level of security and performance.

1.3 Organizations

The rest of this paper is organized as follows. In Section 2, we briefly review some preliminaries including bilinear pairings and mathematical problems involved in our scheme. We describe the definition, security properties and adversary model of certificateless SDVS in Section 3. Then in Section 4, we present our certificateless SDVS scheme concretely. Security analysis of the proposed scheme is discussed in Section 5. A comparison of performance and security with other existing certificateless SDVS schemes is in Section 6. Finally, Section 7 concludes this paper.

2 Preliminaries

In this section, we briefly introduce the concept of bilinear pairings [13, 16] and the complexity assumptions involved in the proposed certificateless SDVS scheme.

Assume $\mathbb{F}_p$ is a finite field in which p is a large prime. Choose randomly $a, b \in_R \mathbb{F}_p$ as two elements to define a curve $\mathbb{E}$. Let $\mathbb{G}$ be an additive cyclic group whose prime order is q, $\mathbb{G}_T$ be a multiplicative cyclic group with the same order and P be a generator of $\mathbb{G}$.

The mapping $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an admissible bilinear pairing with the following properties:

Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, where $P, Q \in \mathbb{G}$ and $a, b \in_R \mathbb{Z}_q^*$.

Non-degeneracy: There exists $P, Q \in \mathbb{G}$ such that $\hat{e}(P, Q) \neq 1$, which is an identity element of $\mathbb{G}_T$.

Computability: There must be an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}$.

Bilinear Diffie-Hellman Problem (BDHP): Given a random instance $(P, aP, bP, cP) \in \mathbb{G}$ for unknown $a, b, c \in \mathbb{Z}_q^*$, it is difficult to compute $\hat{e}(P, P)^{abc}$.

Elliptic Curve Discrete Logarithm Problem (ECDLP): Given two random points $P, Q \in \mathbb{G}$, it is difficult to compute an integer $a \in \mathbb{Z}_q^*$ to satisfy $Q = aP$.

3 Model for the Proposed Certificateless SDVS

3.1 Definition of the Certificateless SDVS

There are two entities in a certificateless SDVS scheme, the real signer Alice and the designated verifier Bob and a certificateless SDVS scheme consists of eight algorithms which are shown below.

Setup: This algorithm takes the security parameter k to output the system parameter sp and the master key s.

Partial-Private-Key-Extract: Given the master key s and the entity's identity id, the KGC generates the entity's partial private key S_{id}.

Set-Secret-Value: The entity chooses randomly a value x_{id} as his/her secret value.

Set-Private-Key: Given the partial private key S_{id} and the secret value x_{id}, the entity outputs his/her full private key sk_{id}.

Set-Public-Key: Given the secret value x_{id}, the public parameter P, this algorithm generates the entity's public key pk_{id}.

Sign: Given the message m, Alice's private key sk_A, Bob's public key pk_B and the system parameter sp, Alice generates the designated verifier signature δ and sends it to Bob.

Verify: Given the message m, Alice's public key pk_A, Bob's private key sk_B, the system parameter sp and the signature δ, Bob outputs $True$ if the signature is correct, otherwise outputs $\perp$.

Transcript-Simulation: Given the message m, Alice's public key pk_A, Bob's private key sk_B and the system parameter sp, Bob generates an indistinguishable designated verifier signature δ'.

3.2 Security Properties of the Certificateless SDVS

1) Correctness: If the signer produces a valid SDVS in the signing phase, it must be accepted in the verifying phase successfully.

2) Unforgeability: Without the private key of the signer or the designated verifier, it is computationally infeasible to forge a valid SDVS for the third party.

3) Source hiding: Given a message-signature pair, the private keys of the signer and the designated verifier, it is computationally infeasible for any polynomial-time distinguisher to determine who is the real signer between the signer and the designated verifier.

4) Non-delegatibity: If a third party is capable of producing a valid signature, he/she must 'know' the private key of the signer or the designated verifier.

Remark: Especially, we present the significance of the non-delegatability property for a designated verifier signature in real applications briefly. As stated previously, the designated verifier signature can be used in software licensing. In order to prevent the software buyers from selling the software they have bought to other people and protect the dealers' benefit, the dealers can produce a designated verifier signature binding with the merchandise to the buyers. In this way, only the actual buyer is able to check the validity of signature, namely, the legality of software. We can utilize the scheme in [11] to realize this real application. The *Signing* phase and the *Verifying* phase can be described as follows.

- Signing: the dealer chooses $r \in_R \mathbb{Z}_q$ and computes $U = rQ_A$, $\sigma = H_2(m, e(S_A, rQ_B))$. Then the signature will be (U, σ).
- Verifying: the buyer checks if $\sigma = H_2(m, e(U, S_B))$ holds or not.

Unfortunately, this scheme can not satisfy the property of non-delegatability because of the common value between the signer and the verifier. The buyer can disclose the common value $e(Q_A, S_B)$ to the third party. Once the third party gets this value, he will be able to check the correction of the equation $\sigma = H_2(m, e(U, S_B))$. Thus, the third party will trust that the software got from the buyer is legal and he will buy it. The non-delegatability can prevent this circumstance from happening perfectly. In the scheme that is equipped with non-delegatability, the common value should not be found.

3.3 Adversary Model of the Certificateless SDVS

There are two types of adversaries proposed by Al-Riyami et al. [1] in certificateless cryptography as follows.

Type 1 Adversary: The adversary can not obtain the master key, namely, the adversary can not obtain the partial private key from the KGC. However, it is capable of replacing the public key of any entity, because there is no public key certificate involved. We can define the attack model as the following game between a challenger C and Type 1 adversary A_1.

Setup: The challenger C firstly takes the security parameter k to generate the system parameter sp and the master key s. C transfers sp to A_1 and keeps the master key s secret meanwhile.

Queries: The adversary A_1 issues the following queries adaptively for polynomially many times:

- Hash queries: Given a hash query for any input, C returns a result to the adversary A_1.
- Partial-Private-Key-Extract queries: Given a partial private key query for any user ID_i, C returns a partial private key S_i for the corresponding user ID_i to the adversary A_1.
- Set-Secret-Value queries: Given a secret value query for any user ID_i, C returns a secret value x_i for the corresponding user ID_i to the adversary A_1.

- Public-Key-Extract queries: Given a public key query for any user ID_i, C returns a public key pk_i for the corresponding user ID_i to the adversary A_1.
- Public-Key-Replacement queries: The adversary A_1 can select a new public key pk_i' for user ID_i to replace the previous public key pk_i. In this way, pk_i' will be the new public key of ID_i.
- Sign queries: Given any message m with signer's identity ID_i and designated verifier's identity ID_j, C returns the corresponding signature δ to the adversary A_1.
- Verify queries: The adversary A_1 can ask for the verification of a message-signature pair (m, δ) with the signer's identity ID_i and verifier's identity ID_j, then C executes the verify algorithm and outputs $True$ if (m, δ) is valid. Otherwise, C outputs $\perp$.

Forgery: Finally, A_1 produces a forged message-signature pair (m^*, δ^*) with signer's identity ID_i and verifier's identity ID_j. The adversary A_1 wins the game if

1) $Verify(m^*, \delta^*, sk_i, pk_j) \rightarrow 1$;
2) A_1 did not issue queries to C on input ID_i and ID_j through Partial-Private-Key-Extract queries, Set-Secret-Value queries or Public-Key-Replacement queries;
3) A_1 did not issue queries to C on input ID_i and ID_j to get the certificateless SDVS on m^* through Sign queries.

Type 2 Adversary: The adversary can obtain the master key, namely, the adversary can generate the entity's partial private key from the KGC. Contrary to Type 1 adversary, this adversary is not capable of replacing the public key of any entity. We can define the attack model as the following game between a challenger C and the Type 2 adversary A_2.

Setup: The challenger C firstly takes the security parameter k to generate the system parameter sp and the master key s. C transfers sp to A_2 and keeps the master key s secret meanwhile.

Queries: The adversary A_2 issues the following queries adaptively for polynomially many times:

- Hash queries: Given a hash query for any input, C returns a result to the adversary A_2.
- Set-Secret-Value queries: Given a secret value query for any user ID_i, C returns a secret value x_i for the corresponding user ID_i to the adversary A_2.
- Public-Key-Extract queries: Given a public key query for any user ID_i, C returns a public key pk_i for the corresponding user ID_i to the adversary A_2.
- Sign queries: Given any message m with signer's identity ID_i and designated verifier's identity ID_j, C returns the corresponding signature δ to the adversary A_2.
- Verify queries: The adversary A_2 can ask for the verification of a message-signature pair (m, δ) with signer's identity ID_i and verifier's identity ID_j, then C executes the verify algorithm and outputs $True$ if (m, δ) is valid. Otherwise, C outputs $\perp$.

Forgery: Finally, A_2 produces a forged message-signature pair (m^*, δ^*) with signer's identity ID_i and verifier's identity ID_j. The adversary A_2 wins the game if

1) $Verify(m^*, \delta^*, sk_i, pk_j) \rightarrow 1$;
2) A_2 did not issue queries to C on input ID_i and ID_j through Partial-Private-Key-Extract queries or Set-Secret-Value queries;
3) A_2 did not issue queries to C on input ID_i and ID_j to get the certificateless SDVS on m^* through Sign queries.

4 Our Proposed Scheme

In this section, we specify our certificateless SDVS scheme which is composed of the following eight algorithms.

Setup: Assume $\mathbb{F}_p$ is a finite field in which p is a large prime. Choose randomly $a, b \in_R \mathbb{F}_p$ as two elements to define a curve $\mathbb{E}$. Let $\mathbb{G}$ be an additive cyclic group whose prime order is q, $\mathbb{G}_T$ be a multiplicative cyclic group with the same order and P be a generator of $\mathbb{G}$. The mapping $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is an admissible pairing. Let $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$, $H_2 : \{0,1\}^* \times \mathbb{G} \rightarrow \mathbb{G}$,$H_3 : \mathbb{G} \times \mathbb{G}_T \rightarrow \mathbb{Z}_q^*$ be three cryptographic hash functions. The system parameter sp is $(\mathbb{F}_p, a, b, P, q, \hat{e}, H_1, H_2, H_3)$, the KGC randomly selects $s \in \mathbb{Z}_q^*$ as master key and keeps it secret.

Partial-Private-Key-Extract: This algorithm accepts an identity $ID_i \in \{0,1\}^*$, $i \in A, B$ and constructs the partial private key for the user as follows:

1) Compute $Q_i = H_1(ID_i)$.
2) Output the partial private key $S_i = sQ_i$.

Set-Secret-Value: This user selects a random $x_i \in \mathbb{Z}_q^*$ and outputs x_i, $i \in A, B$ as his/her secret value. That is, the sender Alice randomly selects $x_A \in \mathbb{Z}_q^*$ and the designated verifier Bob randomly selects $x_B \in \mathbb{Z}_q^*$.

Set-Private-Key: The full private key of Alice and Bob will be $sk_A = (x_A, S_A)$ and $sk_B = (x_B, S_B)$.

Set-Public-Key: This algorithm computes $pk_A = x_A P$ and $pk_B = x_B P$ as Alice and Bob's public keys respectively.

Sign: Assume the message is m, then the signer Alice randomly selects $r, l \in \mathbb{Z}_q^*$ and computes

$$\begin{aligned} A &= lP, \\ C_0 &= rP, \\ C_1 &= H_2(m, A) \\ C &= C_0 + C_1 = (c_x, c_y), \\ v &= l + c_x x_A \\ R &= rpk_B, \\ \sigma &= H_3(R, \hat{e}(S_A, Q_B)). \end{aligned}$$

Finally, the signature δ on message m for the designated verifier Bob is (C, v, σ).

Verify: Once receiving the signature δ, the verifier Bob computes

$$\begin{aligned} A' &= vP - c_x pk_A, \\ C_1' &= H_2(m, A), \\ C_0' &= C - C_1', \\ R' &= x_B C_0', \\ \sigma' &= H_3(R', \hat{e}(Q_A, S_B)). \end{aligned}$$

Bob accepts the signature δ if and only if the equation $\sigma = \sigma'$ holds.

Transcript-Simulation: The verifier Bob can produce a valid signature δ' intended for himself by executing the following operations: Randomly selects $C \in \mathbb{G}$, $v \in \mathbb{Z}_q^*$ and computes

$$\begin{aligned} A &= vP - c_x pk_A, \\ C_1 &= H_2(m, A), \\ C_0 &= C - C_1, \\ R &= x_B C_0, \\ \sigma &= H_3(R, \hat{e}(Q_A, S_B)). \end{aligned}$$

Then the signature δ' is (C, v, σ).

5 Security Analysis

5.1 Correctness

The transcript-simulation algorithm is correct obviously and the correctness of the verifying algorithm is validated as follows:

$$\begin{aligned} A' &= vP - c_x pk_A \\ &= (l + c_x s)P - c_x pk_A \\ &= lP + c_x pk_A - c_x pk_A \\ &= lP \\ &= A. \\ R' &= x_B C_0' \\ &= x_B rP \\ &= rpk_B = R. \end{aligned}$$

5.2 Unforgeability against Type 1 Adversary

Theorem 1. *The proposed certificateless SDVS scheme is existentially unforgeable against Type 1 adversary in the random oracle model under the hardness of BDHP.*

Proof. Assume that A_1 is Type 1 adversary who can forge a valid certificateless SDVS with a non-negligible probability and within the polynomial time t. There exists an algorithm C which treats A_1 as a black box to solve the BDHP with a non-negligible probability. That is, for given a random instance $(P, aP, bP, cP) \in \mathbb{G}$ and for the unknown $a, b, c \in \mathbb{Z}_q^*$, C is able to compute $\hat{e}(P, P)^{abc}$. The game is shown as follows:

Setup: The challenger C firstly takes the security parameter k to generate the system parameter $sp = (\mathbb{F}_p, a, b, P, q, \hat{e}, H_1, H_2, H_3)$ and the master key s. C transfers sp to A_1 and keeps the master key s secret meanwhile.

Queries: The adversary A_1 issues the following queries adaptively for polynomially many times:

- Hash queries to H_1: Suppose that A_1 can send at most q_{H_1} times H_1 queries and C preserve a list $L_{H_1}^{list}$. The list is used to store the tuple of form (ID_i, Q_i, d_i) and set to be empty initially. C responds as follows if A_1 transfers a H_1 query with ID_i.
  1) If $ID_i = ID_A$, return $Q_i = H_1(ID_i) = aP$ to A_1, then append a new tuple $(ID_i, Q_i, \perp)$ to the list $L_{H_1}^{list}$.
  2) Else if $ID_i = ID_B$, return $Q_i = H_1(ID_i) = bP$ to A_1, then append a new tuple $(ID_i, Q_i, \perp)$ to the list $L_{H_1}^{list}$.
  3) Else, return $Q_i = H_1(ID_i) = d_i P$ to A_1, where $d_i \in \mathbb{Z}_q^*$, then append a new tuple (ID_i, Q_i, d_i) to the list $L_{H_1}^{list}$.
- Partial-Private-Key-Extract queries: The challenger C preserves a list L_{ppke}^{list} composed of the tuple of the form (ID_i, D_i, S_i). Once receiving a Partial-Private-Key-Extract query on ID_i, C looks up the tuple (ID_i, Q_i, S_i) from L_{ppke}^{list} and responds as follows:
  1) If $ID_i \neq ID_A$ and $ID_i \neq ID_B$, C looks up the tuple (ID_i, Q_i, d_i) in the list $L_{H_1}^{list}$. If the tuple exists, C returns $S_i = d_i cP$ to A_1. Otherwise, C chooses randomly a number $d_i \in \mathbb{Z}_q^*$, then returns $S_i = d_i cP$ to A_1. Afterwards, C appends (ID_i, Q_i, S_i) to L_{ppke}^{list}.
  2) Else if $ID_i = ID_A$ or $ID_i = ID_B$, C terminates the protocol.
- Public-Key-Extract queries: C preserves a list L_{pk}^{list} composed of the tuple of the form

(ID_i, Q_i, pk_i, x_i). Once A_1 calls a Public-Key-Extract query on ID_i, C looks up the tuple (ID_i, Q_i, pk_i, x_i) from the list L_{pk}^{list} and responds as follows:

1) If L_{pk}^{list} includes (ID_i, Q_i, pk_i, x_i), C returns pk_i to A_1.
2) Else if, C randomly chooses a value $x_i \in \mathbb{Z}_q^*$, computes $pk_i = x_i P$ and returns pk_i to A_1, then appends a new tuple (ID_i, Q_i, pk_i, x_i) to L_{pk}^{list}.

- Set-Secret-Value queries: Once receiving a Set-Secret-Value query on ID_i from A_1, C looks up the tuple (ID_i, Q_i, pk_i, x_i) from the list L_{pk}^{list}. If L_{pk}^{list} includes (ID_i, Q_i, pk_i, x_i), C returns x_i. Otherwise, C performs a Public-Key-Extract query on x_i to produce (ID_i, Q_i, pk_i, x_i), returns the secret value x_i to A_1 and appends the tuple to L_{pk}^{list}.
- Public-Key-Replacement queries: Once receiving a Public-Key-Replacement query on (ID_i, pk_i'), C looks up the tuple (ID_i, Q_i, pk_i, x_i) from the list L_{pk}^{list}. If the L_{pk}^{list} includes (ID_i, Q_i, pk_i, x_i), C sets $pk_i = pk_i'$ and updates the tuple $(ID_i, Q_i, pk_i, x_i = \perp)$. Otherwise, C executes a Public-Key-Extract query to produce (ID_i, Q_i, pk_i, x_i), sets $pk_i = pk_i'$ and updates the tuple $(ID_i, Q_i, pk_i, x_i = \perp)$, Then appends the new tuple $(ID_i, Q_i, pk_i, x_i = \perp)$ to L_{pk}^{list}.
- Hash queries to H_2: C preserves a list H_2^{list} composed of the tuple of the form $(m, A \in \mathbb{G}, C_1)$. Once receiving a Hash queries to H_2 on (m_i, A_i), C executes as follows:
  1) if H_2^{list} includes the tuple (m_i, A_i, C_{1i}), C returns C_{1i} to A_1 as a response.
  2) Otherwise, C randomly chooses $C_{1i} \in \mathbb{G}$, sends it to A_1 and appends (m_i, A_i, C_{1i}) to the list H_2^{list}.
- Hash queries to H_3: C preserves a list H_3^{list} composed of the tuple of the form $(R \in \mathbb{G}, T \in \mathbb{G}_T, \sigma)$. Once receiving a Hash queries to H_3 on (R_i, T_i), C executes as follows:
  1) if H_3^{list} includes the tuple (R_i, T_i, σ_i), C returns σ_i to A_1.
  2) Otherwise, C randomly chooses $\sigma_i \in \mathbb{Z}_q^*$, sends it to A_1 and appends (R_i, T_i, σ_i) to the list H_3^{list}.
- Sign queries: Once receiving a Sign query on input a message m, a signer's identity ID_i and a designated verifier's identity ID_j from A_1, C responds as follows:
  1) If $ID_i \neq ID_A$, or $ID_i \neq ID_B$, C extracts (ID_i, Q_i, S_i) and (ID_i, Q_i, pk_i, x_i) from the list L_{ppke}^{list} and L_{pk}^{list} respectively to get the signer ID_i's private key $(x_i, S_i) = (x_i, d_i(cP))$. C randomly selects $r, l \in \mathbb{Z}_q^*$, computes $A = lP$, $C_0 = rP$, $C_1 = H_2(m, A)$, $C = C_0 + C_1 = (c_x, c_y)$, $v = l + c_x x_i$, $R = rpk_j$, $\sigma = H_3(R, \hat{e}(S_i, Q_j))$ to produce the signature (C, v, σ) and returns it to A_1.
  2) Else if $ID_j \neq ID_A$, or $ID_j \neq ID_B$, C extracts (ID_j, Q_j, S_j) and (ID_j, Q_j, pk_j, x_j) from the list L_{ppke}^{list} and L_{pk}^{list} respectively. C randomly selects $C \in \mathbb{G}, v \in \mathbb{Z}_q^*$, computes $A = vP - c_x pk_i$, $C_1 = H_2(m, A)$, $C_0 = C - C_1$, $R = x_j C_0$, $\sigma = H_3(R, \hat{e}(Q_i, S_j))$ to produce the signature (C, v, σ) and returns it to A_1.
  3) Else, C terminates the protocol.
- Verify queries: Once receiving a Verify query on input a message-signature pair (m, δ), a signer's identity ID_i and a designated verifier's identity ID_j from A_1, C responds as follows:
  1) If $ID_i = ID_A, ID_j = ID_B$ or $ID_i = ID_B, ID_j = ID_A$, C aborts the protocol execution.
  2) Otherwise, C extracts (ID_j, Q_j, S_j) and (ID_j, Q_j, pk_j, x_j) from the list L_{ppke}^{list} and L_{pk}^{list} respectively to get the designated verifier ID_j's private key $(x_j, S_j) = (x_j, d_j(cP))$, then validates the signature through the verify algorithm in our proposed scheme.

Forgery: In the end, A_1 produces a valid certificateless SDVS $\delta = (C^*, v^*, \sigma^*)$ on input a chosen message m^*, a signer's identity ID_i and a designated verifier's identity ID_j. If $(ID_i, ID_j) \neq (ID_A, ID_B)$ or $(ID_i, ID_j) \neq (ID_B, ID_A)$, C aborts the protocol execution and outputs $Fail$. Otherwise, C produces a valid signature $\sigma^* = H_3(R^*, \hat{e}(S_A, Q_B))$ for figuring out $\hat{e}(S_A, Q_B) = \hat{e}(acP, bP) = \hat{e}(P, P)^{abc}$. Thus, the BDHP is resolved. Unfortunately, It is infeasible to address the intractable BDHP by any polynomial time algorithm.

□

5.3 Unforgeability against Type 2 Adversary

Theorem 2. *The proposed certificateless SDVS scheme is existentially unforgeable against the adversaries 2 in the random oracle model under the hardness of ECDLP.*

Proof. Assume that A_2 is Type 2 adversary who can forge a valid certificateless SDVS with a non-negligible probability and within the polynomial time t. There exists an algorithm C which treats A_2 as a black box to solve the ECDLP with a non-negligible probability. That is, for given two random points $P, Q \in \mathbb{G}$, C is able to compute

Table 1: Notation and description of cryptographic operations

Notation	Description
C_P	Pairing operation
C_S	Scalar multiplication operation in $\mathbb{G}$
C_H	Hash operation
C_E	Exponentiation operation
C_I	Inversion operation
C_A	Add operation in $\mathbb{G}$

Table 2: Performance comparison of our scheme with other existing schemes

Schemes	Signature-size	Sign-cost	Verify-cost
Huang et al. [6]	$\|\mathbb{Z}_q^*\|$	$1C_P + 1C_S + 1C_H + 1C_I$	$3C_P + 1C_S + 1C_H$
Chen et al. [3]	$\|\mathbb{Z}_q^*\|$	$1C_P + 1C_S + 1C_H$	$1C_P + 1C_S + 1C_H$
Du et al. [4]	$2\|\mathbb{Z}_q^*\| + \|\mathbb{G}\|$	$3C_S + 1C_H + 1C_E$	$2C_P + 3C_S + 1C_H$
Yang et al. [24]	$2\|\mathbb{G}\|$	$1C_P + 4C_S + 1C_H$	$1C_P + 2C_S + 1C_H$
Hafizul et al. [8]	$2\|\mathbb{G}\|$	$3C_P + 3C_S + 2C_H + 1C_E$	$1C_P + 1C_S + 1C_H + 1C_E$
Ours	$2\|\mathbb{Z}_q^*\| + \|\mathbb{G}\|$	$1C_P + 3C_S + 2C_H$	$1C_P + 3C_S + 2C_H$

Table 3: Security comparison of our scheme with other existing schemes

Schemes	Non-delegatability	Resilience against Type 1 adversary	Resilience against Type 2 adversary
Huang et al. [6]	NO	YES	NO
Chen et al. [3]	NO	YES	YES
Du et al. [4]	NO	YES	YES
Yang et al. [24]	NO	YES	YES
Hafizul et al. [8]	NO	YES	NO
Ours	YES	YES	YES

an integer $a \in \mathbb{Z}_q^*$ to satisfy $Q = aP$. The game is shown as follows:

Setup: The challenger C firstly takes the security parameter k to generate the system parameter $sp = (\mathbb{F}_p, a, b, P, q, \hat{e}, H_1, H_2, H_3)$ and the master key s. C transfers sp to A_2 and keeps the master key s secret meanwhile.

Queries: The adversary A_2 issues the following queries adaptively for polynomially many times:

- Hash queries to H_1: C preserves a list H_1^{list} composed of the tuple of the form (ID_i, Q_i, d_i). Once A_2 issues a Hash queries to H_1 on ID_i, C searches the tuple (ID_i, Q_i, d_i) from the list H_1^{list}. If H_1^{list} includes (ID_i, Q_i, d_i), C returns the previous value Q_i. Otherwise, C randomly chooses a value $d_i \in \mathbb{Z}_q^*$, returns $Q_i = d_iP$ to A_2 and inserts (ID_i, Q_i, d_i) to the list H_1^{list}.
- Public-Key-Extract queries: C preserves a list L_{pk}^{list} composed of the tuple of the form (ID_i, Q_i, pk_i, x_i). Once A_2 issues a Public-Key-Extract query on ID_i, C searches the tuple (ID_i, Q_i, pk_i, x_i) from the list L_{pk}^{list} and executes the following steps.
  1) If L_{pk}^{list} includes (ID_i, Q_i, pk_i, x_i) and
     - If $ID_i \neq ID_A, ID_i \neq ID_B$, the challenger C returns the previous value pk_i.
     - Else if $ID_i = ID_A$ or $ID_i = ID_B$, the challenger C returns $pk_i = aP$ or $pk_i = bP$ as response and appends a new tuple $(ID_i, Q_i, pk_i, \perp)$ to the list L_{pk}^{list}.
  2) Else if there does not exist this tuple, C randomly selects a value $x_i \in \mathbb{Z}_q^*$, returns $pk_i = x_iP$ and inserts a new tuple (ID_i, Q_i, pk_i, x_i) to the list L_{pk}^{list}.
- Set-Secret-Value queries: Once A_2 issues a Set-Secret-Value query on ID_i, C searches (ID_i, Q_i, pk_i, x_i) from the list L_{pk}^{list}.
  1) If L_{pk}^{list} includes (ID_i, Q_i, pk_i, x_i) and
     - If $ID_i \neq ID_A, ID_i \neq ID_B$, the challenger C returns the previous value x_i to A_2.
     - Else if $ID_i = ID_A$ or $ID_i = ID_B$, C terminates the protocol.
  2) Else if there does not exist this tuple, C randomly returns a value $x_i \in \mathbb{Z}_q^*$, computes $pk_i = x_iP$ and inserts a new tuple (ID_i, Q_i, pk_i, x_i) to the list L_{pk}^{list}.
- Hash queries to H_2, Hash queries to H_3, Sign queries, Verify queries: Since these steps are the

same as the corresponding steps in Theorem 1, we do not make those statements again.

Forgery: In the end, A_2 produces a valid certificateless SDVS $\delta = (C^*, v^*, \sigma^*)$ on input a chosen message m^*, a signer's identity ID_i and a designated verifier's identity ID_j. If $(ID_i, ID_j) \neq (ID_A, ID_B)$ or $(ID_i, ID_j) \neq (ID_B, ID_A)$, C aborts the protocol execution and outputs $Fail$. Otherwise, C produces a valid $v^* = l + c_x x_A$ for figuring out $x_A = pk_A/P$. Thus, the ECDLP is resolved. Unfortunately, It is infeasible to address the intractable ECDLP by any polynomial time algorithm.

□

Theorem 3. *The proposed certificateless SDVS scheme is equipped with the property of source hiding in the random oracle model.*

Proof. Given a message-signature pair (m, δ), the signer' private key (x_A, S_A) and the designated verifier' private key (x_B, S_B) used in the proposed construction, a third party can not distinguish who is the real signer. The reason is that the following two equations always hold.

$$\begin{aligned} R &= rpk_B = x_B C_0, \\ \sigma &= H_3(R, \hat{e}(S_A, Q_B)) \\ &= H_3(R, \hat{e}(Q_A, S_B)). \end{aligned}$$

□

Theorem 4. *The proposed certificateless SDVS scheme is equipped with the resistance against delegatability attack in the random oracle model.*

Proof. In our scheme, the signer's secret value x_A and the verifier's secret value x_B are used solely in the signing phase and the verifying phase such that there does not exist disclosing of the common value $(x_A x_B P)$. Although another common value $\hat{e}(S_A, Q_B)$ is possible to be transferred to the third party, the fact that the third party can not figure out the value $v = l + c_x x_A$ prevents the third party from creating a valid signature. The delegatability attack only could happen when the secret value x_A and the common value $\hat{e}(S_A, Q_B)$ are disclosed concurrently, but the probability is negligible. In this situation, it is infeasible for the delegatability attacker to defeat our scheme. □

6 Comparison

In this section, we present a comparison of the proposed scheme with other existing certificateless SDVS in terms of performance and security. The notations and the corresponding descriptions of cryptographic operations are shown in Table 1. Table 2 is for performance comparison and Table 3 is for security comparison. We assume that the bit length of element in $\mathbb{G}$ is $|\mathbb{G}|$ and the bit length of element in $\mathbb{Z}_q^*$ is $|\mathbb{Z}_q^*|$. The length of the signature in our scheme is equal to the length in [4], so it is acceptable. Among the cryptographic operations listed in Table 1, pairing operation is recognized as the most time-consuming operation and add operation in $\mathbb{G}$ can be neglected because of its low computational cost. By contrast with the existing schemes, we can see our scheme only requires one pairing operation no matter in the signing phase or in the verifying phase such that the computing consumption of our scheme is at a very low level. Table 3 shows that only our scheme is able to satisfy the three important properties in certificate SDVS scheme simultaneously. In summary, our scheme is relatively efficient and provably secure among the existing schemes.

7 Conclusions

This paper proposes the first certificateless SDVS scheme with non-delegatability. The proposed scheme extends the Schnorr digital signature to a certificateless SDVS. We provide the security proof of the proposed scheme on the basic properties of SDVS. We also prove that our scheme can resist the two types of adversaries in certificateless cryptography. In addition, the comparison with other existing certificateless SDVS shows that our scheme has a higher level of efficiency and security.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China under Grant Nos. 61003230, 61370026, 61300191 and 61272029.

References

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *9th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'03)*, pp. 452–473, Taipei, Taiwan, Nov.–Dec. 2003.

[2] M. H. Au, Y. Mu, J. Chen, D. S. Wong, J. K. Liu, and G. Yang, "Malicious kgc attacks in certificateless cryptography," in *2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS'07)*, pp. 302–311, Singapore, Mar. 2007.

[3] H. Chen, R. Song, F. Zhang, and F. Song, "An efficient certificateless short designated verifier signature scheme," in *4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08)*, pp. 1–6, Dalian, China, Oct. 2008.

[4] H. Du and Q. Wen, "Efficient certificateless designated verifier signatures and proxy signatures," *Chinese Journal of Electronics*, vol. 18, no. 1, pp. 95–100, 2009.

[5] Q. Huang, G. Yang, D. S. Wong, and W. Susilo, "Identity-based strong designated verifier signature revisited," *Journal of Systems and Software*, vol. 84, no. 1, pp. 120–129, 2011.

[6] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "Certificateless designated verifier signature schemes," in *20th International Conference on Advanced Information Networking and Applications (AINA'06)*, pp. 15–19, Vienna, Austria, Apr. 2006.

[7] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "Short (identity-based) strong designated verifier signature schemes," in *7th International Conference on Information Security Practice and Experience (ISPEC'11)*, pp. 214–225, Hangzhou, China, May–Jun. 2006.

[8] S. H. Islam and G. P. Biswas, "Provably secure certificateless strong designated verifier signature scheme based on elliptic curve bilinear pairings," *Journal of King Saud University-Computer and Information Sciences*, vol. 25, no. 1, pp. 51–61, 2013.

[9] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," in *International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'96)*, pp. 143–154, Saragossa, Spain, May 1996.

[10] P. K. Kancharla, S. Gummadidala, and A. Saxena, "Identity based strong designated verifier signature scheme," *Informatica*, vol. 18, no. 2, pp. 239–252, 2007.

[11] B. Kang, C. Boyd, and E. Dawson, "Identity-based strong designated verifier signature schemes: attacks and new construction," *Computers and Electrical Engineering*, vol. 35, no. 1, pp. 49–53, 2009.

[12] B. Kang, C. Boyd, and E. D. Dawson, "A novel identity-based strong designated verifier signature scheme," *Journal of Systems and Software*, vol. 82, no. 2, pp. 270–273, 2009.

[13] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[14] C. T. Li and Hwang M. S. A., "secure and anonymous electronic voting scheme based on key exchange protocol," *International Journal of Security and Its Applications*, vol. 7, no. 1, pp. 59–70, 2009.

[15] H. Lipmaa, G. Wang, and F. Bao, "Designated verifier signature schemes: attacks, new security notions and a new construction," in *32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, pp. 459–471, Lisbon, Portugal, July 2005.

[16] L. Liu and Z. Cao, "A note on efficient algorithms for secure outsourcing of bilinear pairings," *International Journal of Electronics and Information Engineering*, vol. 5, no. 1, pp. 30–36, 2016.

[17] T. Liu, X. Wang, and X. Ding, "security analysis and improvement of certificateless strong designated verifier signature scheme," *Computer Science*, vol. 40, no. 7, pp. 126–128, 2013 (in Chinese).

[18] C. Y. Ng, W. Susilo, and Y. Mu, "Universal designated multi verifier signature schemes," in *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, pp. 305–309, Fukuoka, Japan, July 2005.

[19] S. Saeednia, S. Kremer, and O. Markowitch, "An efficient strong designated verifier signature scheme," in *6th International Conference on Information Security and Cryptology (ICISC'03)*, pp. 40–54, Seoul, Korea, Nov. 2004.

[20] K. A. Shim, "On delegatability of designated verifier signature schemes," *Information Sciences*, vol. 281, pp. 365–372, 2014.

[21] W. Susilo, F. Zhang, and Y. Mu, "Identity-based strong designated verifier signature schemes," in *9th Australasian Conference on Information Security and Privacy (ACISP'04)*, pp. 313–324, Sydney, Australia, July 2004.

[22] H. Tian, X. Chen, Z. Jiang, and Y. Du, "Non-delegatable strong designated verifier signature on elliptic curves," in *14th International Conference on Information Security and Cryptology (ICISC'11)*, pp. 219–234, Seoul, Korea, Nov.–Dec. 2012.

[23] H. Tian, X. Chen, F. Zhang, B. Wei, Z. Jiang, and Y. Liu, "A non-delegatable strong designated verifier signature in id-based setting for mobile environment," *Mathematical and Computer Modelling*, vol. 58, no. 5, pp. 1289–1300, 2013.

[24] B. Yang, Z. Hu, and Z. Xiao, "Efficient certificateless strong designated verifier signature scheme," in *International Conference on Computational Intelligence and Security (CIS'09)*, vol. 1, pp. 432–436, Beijing, China, Dec. 2009.

[25] J. Zhang and J. Mao, "A novel id-based designated verifier signature scheme," *Information Sciences*, vol. 178, no. 3, pp. 766–773, 2008.

[26] M. Zhang, T. Takagi, Y. A. N. G. Bo, and L. I. Fagen, "Cryptanalysis of strong designated verifier signature scheme with non-delegatability and non-transferability," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, no. 1, pp. 259–262, 2012.

Biography

Yang Chen is a teaching assistant of Sichuan Aerospace Vocational College. He received his M.S. degree from University of Electronic Science and Technology of China (UESTC). His research covers cryptography and information security.

Yang Zhao is an associate professor at UESTC. His research interests are in the area of networking security and e-commerce protocol.

Hu Xiong is an associate professor at UESTC. He received his Ph.D degree from UESTC in 2009. His research interests include: cryptography and network security.

Feng Yue received his B.S. degree in the School of International Education, Henan University of Science and Technology (HAUST) in 2008. He is currently pursing his M.S. degree in the School of Computer Science and Engineering, UESTC. His research interests include: cryptography and information security.