Hindawi
Security and Communication Networks
Volume 2017, Article ID 7921782, 9 pages
https://doi.org/10.1155/2017/7921782

Research Article

Strong Designated Verifier Signature Schemes with Undeniable Property and Their Applications

Xiaoming Hu,[1] Wenan Tan,[1] Huajie Xu,[2] Jian Wang,[1] and Chuang Ma[1]

[1]*College of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China*
[2]*School of Computer and Electronic Information, Guangxi University, Nanning 530004, China*

Correspondence should be addressed to Xiaoming Hu; xmhu@sspu.edu.cn

Received 1 August 2016; Revised 21 November 2016; Accepted 22 December 2016; Published 24 January 2017

Academic Editor: Muhammad Khurram Khan

Most of the strong designated verifier signature (SDVS) schemes cannot tell the real signature generator when the signer and the designated verifier dispute on a signature. In other words, most of the SDVS schemes do not have the undeniability property. In this paper, we propose two SDVS schemes which hold the undeniability property, namely, strong designated verifier signature with undeniability property (SDVSUP). Our two schemes are called SDVSUP-1 and SDVSUP-2. In our two SDVSUP schemes, the signer not only can designate a verifier but also can designate an arbiter who can judge the signature when the signer and the designated verifier dispute on the signature. What is more, the judgment procedure can be performed by the arbiter alone without help from the signer or the designated verifier, which increases the judgment efficiency and reduces the complexity of signature confirmation. We also demonstrate a real instance of applying our SDVSUP scheme to electronic bidding system.

1. Introduction

In traditional digital signature (TDS), anyone who knows the public key of the signer can verify the validity of a signature. However, the public verification of TDS is not a desirable property in some applications. For example, the owner of some privacy information such as a health report from hospital or a bill from company and so on wishes that the signature on these privacy information can only be verified by himself. There are some solutions to this problem. One of them is to use the undeniable signature which was proposed first by Chaum and Antwerpen [1, 2]. In undeniable signature (US) [3–7], the signature verification needs the help from the signer. In other words, the validity verification of a signature is an interactive proof between the signer and the verifier which leads to the inefficiency and infeasibility if the signer rejects to cooperate. Another solution is to use the designated verifier signature (DVS) which was proposed first by Jakobsson et al. [8]. In DVS, the signer can designate a person as the signature verifier called designated verifier who can convince the signature to be generated by the signer. But the designated verifier cannot transfer the conviction to any third party since the designated verifier can generate a indistinguishable signature with the signer. This is called nontransferability. Therefore, though a signature is publicly verifiable in DVS but no one can tell that the signature is generated by the signer or the designated verifier. Jakobsson et al. also proposed a variant of DVS called strong designated verifier signature (SDVS) [8]. In SDVS, the signature verification needs the private key of the designated verifier. Thus, no one other than the signer and the designated verifier can verify the validity of signature which further protects the privacy information of the signer.

However, if the signer and the designated verifier dispute on a signature, no one can tell the real generator of the signature either the signer or the designated verifier. Yang et al. [9, 10] gave an instance on this situation. In an electronic bidding system, some companies use SDVS to submit their prices to the institution for a project. Using the SDVS, the institution can confirm the submission but cannot transfer the submission to other companies for lower price since the institution also can generate an indistinguishable submission with the company. But if the winning company denies the submission due to some reasons, such as economic crisis,

bankrupt, and even malicious competition. The institution can do nothing on it. This is undesirable to the institution. However, in almost all SDVS schemes [11–17] proposed till now this problem exists. Namely, these SDVS schemes have no undeniability property. Without undeniability property, SDVS is like more a message authentication code rather than a digital signature [9, 10].

1.1. Related Work. Jakobsson et al. first proposed the concept of DVS and presented a DVS scheme which was based on trapdoor commitments [8]. In Jakobsson et al.'s DVS scheme, a signature generated by the signer with the form of $s = m^{x_S}$ while s was a random element in the signature generated by the designated verifier. Therefore, with the help from the signer, a person could distinguish the signature by an interactive proof between the signer and this person. So, Jakobsson et al.'s DVS scheme held the undeniability property. However, Jakobsson et al. did not explain the property explicitly and consider it as a necessary property. What is more, Lipmaa et al. [18] showed that Jakobsson et al.'s DVS scheme was not undeniable since the signer could construct a valid signature where s was a random element which made the third party confirm the signature from the designated verifier. Lipmaa et al. also proposed a DVS scheme based on Decisional Diffie-Hellman problem. However, their DVS scheme yet did not have the undeniability property.

In order to protect the identity of the signer further, Jakobsson et al.'s [8] extended DVS to present the concept of SDVS. In Jakobsson et al.'s SDVS scheme, the designated verifier must use the private key of himself to verify the validity of the signature. From then, many SDVS were proposed [15, 19–21]. Some other variants of DVS included universal designated verifier signature (UDVS) [7, 22, 23], in which the owner of the standard signature could designate any third party as the designated verifier, identity-based designated verifier signature (IBDVS) [13, 16, 24], in which the private keys of the signer and the designated verifier were generated by the Key Generator Center (KGC), and so on.

In 2012, Yang et al. [9, 10] proposed an SDVS scheme with the undeniability property based on Chameleon hash function [25]. In their SDVS scheme, when the signer and the designated verifier disputed on a signature, the signer confirmed a signature (r, s, ρ, h) if the following two situations held: (1) the signer could find $\widehat{r}$ to hold $r = H(\widehat{r})$, where r was one component of the signature and $\widehat{r}$ was the preimage of r and was stored by the signer in advance; (2) the signer could find an original signature (r', s', ρ', h') of (r, s, ρ, h) where $r' \neq r$, $s' \neq s$, $\rho' = \rho$, $h' = h$, and (r', s', ρ', h') was stored by the signer in advance. Thus, the signer needed to store all original signature data in order to confirm the signature later which added a large storage cost. What is more, anyone could distinguish a signature by the above similar method as the signer, that is, collecting and storing all signature data. And the confirmation procedure of signature was performed only by the signer alone. It was unfair to the designated verifier. What is more, if the signer did not want to cooperate for some reasons, the confirmation procedure could not continue and was forced to stop.

1.2. Our Work. To our knowledge, Jakobsson et al.'s SDVS scheme [8] and Yang et al.'s SDVS scheme [9, 10] are only two SDVS schemes with undeniability property. However, in the two SDVS schemes, it needs a complex judgment procedure when the signer and the designated verifier dispute on a signature. What is more, the judgment needs the help from the signer. In other words, the judgment is an interactive procedure between the signer and the judger. If the signer rejects to cooperate, the judgment procedure cannot be continued and must be stopped. In our work, we propose two SDVS schemes which can solve the above problem. In other words, in our SDVS schemes, the judger or the arbiter can alone complete the judgment: who generates the signature? Either the signer or the designated verifier does. We also make a comparison between our schemes and other similar schemes in terms of computational cost, signature size, and other aspects. At the same time, we present one application instance of our schemes in the electronic bidding system.

The remainder of this paper is organized as follows. In Section 2, some preliminaries are given including Computational Diffie-Hellman problem and assumption, the concept of SDVS, and the security properties of SDVS. In Section 3, two SDVSUP schemes are proposed. The security analysis of two SDVSUP schemes and the comparison are presented. Section 4 concludes this paper.

2. Preliminaries

2.1. Computational Diffie-Hellman (CDH) Problem and CDH Assumption. Let p and q be two large primes which hold $p = 2q + 1$. Let Z_q be a subgroup of Z_p^* with the prime order q and a generator g. Given (g, g^a, g^b) where a and b belong to Z_q^* are two unknown elements, the CDH problem is to compute g^{ab}.

The CDH assumption (t, ϵ) holds in Z_p^* if there is not any algorithm A which can solve the CDH problem with running time at most t and the probability at least ϵ.

2.2. Strong Designated Verifier Signature. A strong designated verifier signature (SDVS) consists of four algorithms, including System Setup, Key Generate, Signature Generate, and Signature Verify.

System Setup (SetSDV). Inputting 1^k where k is a security parameter, the SetSDV algorithm outputs the system parameter *params* and publishes *params* publicly.

Key Generate (KeySDV). Inputting the system parameter *params*, the KeySDV algorithm outputs the public and private key pair (T_s, t_s) of the signer S, the one (T_v, t_v) of the designated verifier V, and the one (T_a, t_a) of the arbiter A.

Signature Generate (SigSDV). Inputting *params*, the public keys of S, V, and A, the private key t_s of S, and a message m, the SigSDV algorithm outputs a signature σ on m.

Signature Verify (VerSDV). Inputting *params*, the public keys of S, V, and A, the private key t_v of V, and a signature σ on a

message m, the VerSDV algorithm outputs "Accept" if σ is a valid signature or "Reject."

If one can verify a signature without the private key t_v of V, then it is called designated verifier signature (DVS) not strong DVS. Namely, inputting *params*, the public keys of S, V, and A and a signature σ on a message m, the VerSDV algorithm outputs "Accept" if σ is a valid signature or "Reject."

A secure strong designated verifier signature with undeniable property (SDVSUP) should hold unforgeability, computationally nontransferability, and undeniability.

2.3. Unforgeability. The unforgeability of an SDVSUP scheme is defined by the following game between the challenger C and an adversary R. The game includes three stages: setup, query, and output.

Setup. The challenger C creates the public system parameter *params* and the public/private key pair (T_s, t_s) of the signer S, the one (T_v, t_v) of the designated verifier V, and the one (T_a, t_a) of the arbiter A. Then, send *params* and (T_s, T_v, T_a) to the adversary R.

Query. Next, R makes the following oracle queries.

(1) Signing Query: R submits a message m to request a signature on m; C generates a valid signature σ on m and returns σ to R.

(2) Verifying Query: R submits a signature σ on a message m; C returns "True" to R if the signature σ is valid. Otherwise, it returns "False" to R.

Output. Finally, R outputs a forged signature σ^* on a message m^*. R wins the above game if

(1) σ^* is a valid signature on m^*,

(2) m^* has never been queried to Signing Query.

An SDVSUP scheme is (t, ϵ, q_s, q_v) unforgeable if no adversary R can win the above game with the time at most t, the probability at least ϵ, making at most q_s signing queries, and making at most q_v verifying queries.

2.4. Nontransferability. According to the work of [18, 22], the nontransferability of SDVSUP can be classified into two types: computational nontransferability and perfect nontransferability. Based on the concept of nontransferability for SDVS given by [18, 22], we add a participator called arbiter A into the original definition to present a description of nontransferability for SDVSUP.

An SDVSUP scheme is computationally nontransferable if given a pair of message and signature (m, σ); it is infeasible for any probabilistic polynomial-time (PPT) algorithm to distinguish that the signature σ is generated by the signer S or the designated verifier V without the knowledge of the secret key of the signer S, the secret key of the designated verifier V, and the secret key of the arbiter A.

An SDVSUP scheme is perfectly nontransferable if one cannot distinguish the signature σ from the signer or the designated verifier even if one knows the secret keys of the signer S, the designated verifier V, and the arbiter A.

Next, we give a definition of computationally nontransferable for SDVSUP scheme. An SDVSUP scheme is computationally nontransferable if there exists a PPT algorithm: Simulate Signature (SimSDV) in which the designated verifier V can use SimSDV to simulate a signature σ_1. σ_1 is indistinguishable from the real signature which is generated by the signer S without knowing the secret key t_s of S, the secret key t_v of V, and the secret key t_a of A. In other words, there is not any PPT distinguisher B that is inputting the public key T_s of S, the public key T_v of V, the public key T_a of the arbiter A, and a signature σ_x to tell the signature σ_x from S or V with a nonnegligible probability ϵ, namely,

$$\Pr\left[\begin{array}{c|c} x = x' & \begin{array}{c} (T_i, t_i) \longleftarrow \mathrm{KeySDV}\left(1^k\right),\ i \in \{s, v, a\}, \\ \sigma_0 \longleftarrow \mathrm{SigSDV}\left(T_s, T_v, T_a, t_s, m\right), \\ \sigma_1 \longleftarrow \mathrm{SimSDV}\left(T_s, T_v, T_a, t_v, m\right), \\ x \longleftarrow_R \{0, 1\}, \\ x' \longleftarrow B\left(T_s, T_v, T_a, \sigma_x\right) \end{array} \end{array}\right] = \epsilon. \tag{1}$$

Similarly, we can define the perfectly nontransferable of SDVSUP scheme with changing the inputting of B into the public/private key (T_s, t_s) of S, the public/private key (T_v, t_v) of V, the public/private key (T_a, t_a) of the arbiter A, and a signature σ_x.

Since there is not any trapdoor information that can be used by the arbiter A even if S and V are in perfect nontransferability, an SDVSUP scheme only holds the computational nontransferability not perfect nontransferability [18].

2.5. Undeniability. An SDVSUP scheme holds the undeniability property if there exists a PPT algorithm: Arbitrate Signature (ArbSDV) with inputting the signature σ on m, the public keys of the signer S and the designated verifier V, and the private key of the arbiter A; the ArbSDV outputs "S" if the signature is generated by the signer S or returns "V" that denotes the signature from the designated verifier V; that is,

$$N \longleftarrow \mathrm{ArbSDV}\left(T_s, T_v, t_a, \sigma\right), \quad N \in \{S, V\}. \tag{2}$$

3. The Proposed Strong Designated Verifier Signature Schemes with Undeniable Property

In this section, we provide two strong designated verifier signature schemes with undeniable property. The first one is called SDVSUP-1 scheme and the another is called SDVSUP-2 scheme.

3.1. The Proposed SDVSUP-1 Scheme. Based on Jakobsson et al.'s scheme [8], we propose a new strong designated verifier signature scheme with undeniable property (SDVSUP-1 scheme). In our SDVSUP-1 scheme, there exists three participators: the signer S, the designated verifier V, and the

arbiter A. Our SDVSUP-1 scheme performs according to the following process.

System Setup (SetSDV). Let p and q be two large primes which hold $p = 2q + 1$. Let Z_q be a subgroup of Z_p^* with the prime order q and a generator g. Define three hash functions which hold F_1: $\{0,1\}^* \to Z_q^*$, F_2: $\{0,1\}^* \times Z_q^* \to Z_q^*$, and F_3: $(Z_q^*)^5 \times \{0,1\}^* \to Z_q^*$. Then, the system parameters are $L = (p, q, g, F_1, F_2, \text{and } F_3)$.

Key Generate (KeySDV). The signer S selects randomly two numbers $t_{s,1}$ and $t_{s,2} \in Z_q^*$ as the private keys of S. And compute $T_{s,1} = g^{t_{s,1}} \bmod p$ and $T_{s,2} = g^{t_{s,1}} \bmod p$ as its public keys. Similarly, the designated verifier V generates its public keys $(T_{v,1}, T_{v,2})$ and private keys $(t_{v,1}, t_{v,2})$, where $t_{v,1}$ and $t_{v,2} \in Z_q^*$ are two random numbers, and $T_{v,1} = g^{t_{v,1}} \bmod p$; $T_{v,2} = g^{t_{v,2}} \bmod p$. The arbiter A generates its public keys $(T_{a,1}, T_{a,2})$ and private keys $(t_{a,1}, t_{a,2})$, where $t_{a,1}$ and $t_{a,2} \in Z_q^*$ are two random numbers, and $T_{a,1} = g^{t_{a,1}} \bmod p$ and $T_{a,2} = g^{t_{a,2}} \bmod p$.

Signature Generate (SigSDV). The signer S constructs a signature on a message m as follows. S selects randomly $k_1, k_2, k_3 \in Z_q^*$ and computes

$$\begin{aligned} K_1 &= T_{a,1}^{k_1} \bmod p. \\ K_2 &= g^{k_1} \bmod p \\ K_3 &= g^{k_2} T_{v,1}^{k_3} \bmod p \\ M &= T_{a,1}^{F_1(m)t_{s,1} + F_2(m,g)t_{s,2}} \bmod p \\ M_1 &= T_{v,1}^{t_{s,1}} \bmod p \\ h &= F_3(K_1, K_2, K_3, M, M_1, m) \\ r &= k_1 + (F(m)\, t_{s,1} + F(m \parallel g)\, t_{s,2})(h + k_2) \bmod q. \end{aligned} \tag{3}$$

The final signature on the message m is $\sigma = (k_2, k_3, h, r, M)$.

Signature Verify (VerSDV). The designated verifier V checks the validity of a signature $\sigma = (k_2, k_3, h, r, M)$ on message m as follows. V computes

$$\begin{aligned} K_1' &= T_{a,1}^r M^{-(h+k_2)} \bmod p \\ K_2' &= g^r \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-(h+k_2)} \bmod p \\ K_3' &= g^{k_2} T_{v,1}^{k_3} \bmod p \\ M_1' &= T_{v,1}^{t_{s,1}} \bmod p \\ h' &= F_3\left(K_1', K_2', K_3', M, M_1', m\right). \end{aligned} \tag{4}$$

If $h = h'$, then V accepts the signature or rejects it.

3.2. Correctness of SDVSUP-1 Scheme. The above signature generated by S is correct because

$$\begin{aligned} &T_{a,1}^r M^{-(h+k_2)} \bmod p \\ &\quad = T_{a,1}^{k_1 + (F_1(m)t_{s,1} + F_2(m,g)t_{s,2})(h+k_2)} M^{-(h+k_2)} \bmod p \\ &\quad = T_{a,1}^{k_1} T_{a,1}^{(F_1(m)t_{s,1} + F_2(m,g)t_{s,2})(h+k_2)} \\ &\quad \cdot \left(T_{a,1}^{F_1(m)t_{s,1} + F_2(m,g)t_{s,2}}\right)^{-(h+k_2)} \bmod p = T_{a,1}^{k_1} \bmod p. \\ &g^r \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-(h+k_2)} \bmod p \\ &\quad = g^{k_1 + (F_1(m)t_{s,1} + F_2(m,g)t_{s,2})(h+k_2)} \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-(h+k_2)} \\ &\quad \cdot \bmod p \\ &\quad = g^{k_1} g^{(F_1(m)t_{s,1} + F_2(m,g)t_{s,2})(h+k_2)} \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-(h+k_2)} \\ &\quad \cdot \bmod p = g^{k_1} \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{(h+k_2)} \\ &\quad \cdot \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-(h+k_2)} \bmod p = g^{k_1} \bmod p. \\ &T_{v,1}^{t_{s,1}} = T_{s,1}^{t_{v,1}} \bmod p. \end{aligned} \tag{5}$$

The above signature simulated by V is correct because

$$\begin{aligned} &T_{a,1}^r M^{-(h+k_2)} \bmod p = T_{a,1}^r M^{-(h+x_1-h)} \bmod p \\ &\quad = T_{a,1}^r M^{-x_1} \bmod p. \\ &g^r \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-(h+k_2)} \bmod p \\ &\quad = g^r \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-(h+x_1-h)} \bmod p \\ &\quad = g^r \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-x_1} \bmod p. \\ &g^{k_2} T_{v,1}^{k_3} \bmod p = g^{x_1 - h} T_{v,1}^{(x_2 - k_2)t_{v,1}^{-1}} \bmod p \\ &\quad = g^{x_1 - h} g^{(x_2 - k_2)} \bmod p = g^{x_1 - h} g^{x_2 - (x_1 - h)} \bmod p \\ &\quad = g^{x_2} \bmod p. \end{aligned} \tag{6}$$

3.3. Security Analysis of SDVSUP-1 Scheme

Theorem 1. *If the CDH assumption $(t_{cdh}, \epsilon_{cdh})$ holds, then our proposed SDVSUP-1 scheme is $(t_{sdv1}, q_{f_1}, q_{f_2}, q_{f_3}, q_s, q_v, \text{and } \epsilon_{sdv1})$ unforgeable, where*

$$\begin{aligned} t_{cdh} &\approx t_{sdv1} + \tau_{exp}(6q_s + 6q_v + 7) \\ &\quad + 4\tau_{mul}(q_s + q_v + 1), \\ \epsilon_{cdh} &\geq \epsilon_{sdv1} - \frac{q_{F_3} q_s}{q} - \frac{2}{q}. \end{aligned} \tag{7}$$

And τ_{exp} is one exponent operation in Z_p^* and τ_{mul} is one multiplication operation in Z_p^*. q_{f_1}, q_{f_2}, q_{f_3}, q_s, and q_v

denote, respectively, that the adversary R is allowed to make at most q_{f_1} F_1 queries, q_{f_2} F_2 queries, q_{f_3} F_3 queries, q_s signing queries, and q_v verifying queries.

Proof. Given a CDH problem instance (g, g^x, g^y), the aim of challenger C is to obtain g^{xy}. Next, C performs the following process with the adversary R.

Setup. C selects randomly $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6 \in Z_q^*$ and sets $T_{s,1} = g^{x\lambda_1} \bmod p$, $T_{s,2} = g^{\lambda_2} \bmod p$, $T_{v,1} = g^{y\lambda_3} \bmod p$, $T_{v,2} = g^{\lambda_4} \bmod p$, $T_{a,1} = g^{\lambda_5} \bmod p$, and $T_{a,2} = g^{\lambda_6} \bmod p$. C publishes the system parameters $L = (p, q, g, F_1, F_2, F_3)$ and the public keys $(T_{s,1}, T_{s,2})$, $(T_{v,1}, T_{v,2})$, and $(T_{a,1}, T_{a,2})$.

Query. The adversary R makes the following queries to C.

Random Oracle Query

(i) When R asks a query on F_1 oracle with inputting m_i, C searches m_i in table T_1 that is empty initially. If there exists a tuple $(m_i, l_{1,i})$ in T_1, C returns $l_{1,i}$ to R as the value of $F_1(m_i)$. Otherwise, C selects randomly $l_{1,i} \in Z_q^*$ and records $(m_i, l_{1,i})$ in T_1 and returns $l_{1,i}$ to R.

(ii) When R asks a query on F_2 oracle with inputting (m_i, g), C searches (m_i, g) in table T_2 that is empty initially. If there exists a tuple $(m_i, g, l_{2,i})$ in T_2, C returns $l_{2,i}$ to R as the value of $F_2(m_i, g)$. Otherwise, C selects randomly $l_{2,i} \in Z_q^*$ and records $(m_i, g, l_{2,i})$ in T_2 and returns $l_{2,i}$ to R.

(iii) When R asks a query on F_3 oracle with inputting $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, M_{1,i}, m_i)$, C searches $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, *, m_i)$ in table T_3 that is empty initially. If there exists a tuple $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, *, m_i, l_{3,i})$ in T_3, C returns $l_{3,i}$ to R as the value of $F_2(K_{1,i}, K_{2,i}, K_{3,i}, M_i, M_{1,i}, m_i)$. Otherwise, C selects randomly $l_{3,i} \in Z_q^*$ and records $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, M_{1,i}, m_i, l_{3,i})$ in T_3 and returns $l_{3,i}$ to R.

Signing Query

(i) When R asks a signature query with inputting m_i, C searches m_i in table T_4 that is empty initially. If there exists a tuple $(m_i, k_{2,i}, k_{3,i}, h_i, r_i, M_i, \perp)$ in T_4, C returns $(k_{2,i}, k_{3,i}, h_i, r_i, M_i)$ to R as the signature on message m_i. Otherwise, C searches T_1 on m_i and T_2 on (m_i, g). If m_i and (m_i, g) have not existed in T_1 and T_2, then C performs the above F_1 oracle and F_2 oracle to obtain $l_{1,i}$ and $l_{2,i}$.

Then, C chooses randomly $r_i, k_{2,i}, k_{3,i}, l_{3,i} \in Z_q^*$, and $M_i \in Z_p^*$. Compute

$$\begin{aligned} K_{1,i} &= T_{a,1}^{r_i} M_i^{-(l_{3,i}+k_{2,i})} \bmod p \\ K_{2,i} &= g^{r_i} \left(T_{s,1}^{l_{1,i}} T_{s,2}^{l_{2,i}}\right)^{-(l_{3,i}+k_{2,i})} \bmod p \\ K_{3,i} &= g^{k_{2,i}} T_{v,1}^{k_{3,i}} \bmod p \end{aligned} \tag{8}$$

and set $F_3(K_{1,i}, K_{2,i}, K_{3,i}, M_i, \perp, m_i) = l_{3,i}$.

If there exists a tuple $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, M_{1,i}, m_i, l'_{3,i})$ in T_3 and $l'_{3,i} \neq l_{3,i}$, then C fails and aborts. Otherwise, C records $(m_i, k_{2,i}, k_{3,i}, h_i, r_i, M_i)$ in T_4 and returns $(k_{2,i}, k_{3,i}, h_i, r_i, M_i)$ to R. And record $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, \perp, m_i, l_{3,i})$ in T_3. The probability of failure for C is $q_{F_3} q_s / q$.

Verifying Query

(i) When R asks a signature verification query with $\sigma = (k_{2,i}, k_{3,i}, h_i, r_i, M_i)$ on the message m_i, C searches $(k_{2,i}, k_{3,i}, h_i, r_i, M_i)$ in table T_4. If there exists a tuple $(m_i, k_{2,i}, k_{3,i}, h_i, r_i, M_i)$ in T_4, C returns "true." Otherwise, C computes

$$\begin{aligned} K_{1,i} &= T_{a,1}^{r_i} M^{-(h_i+k_{2,i})} \bmod p \\ K_{2,i} &= g^{r_i} \left(T_{s,1}^{F_1(m_i)} T_{s,2}^{F_2(m_i,g)}\right)^{-(h_i+k_{2,i})} \bmod p \\ K_{3,i} &= g^{k_{2,i}} T_{v,1}^{k_{3,i}} \bmod p. \end{aligned} \tag{9}$$

Then C searches $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, *, m_i, h_i)$ in T_3. If $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, *, m_i, h_i)$ exits in T_3, then C returns *false*. Otherwise, $(K_{1,i}, K_{2,i}, K_{3,i}, M_i, *, m_i, h_i)$ has not existed in T_3; then C outputs "false" and aborts. The probability which $\sigma = (k_{2,i}, k_{3,i}, h_i, r_i, M_i)$ is a valid signature and did not make a F_3 query is $1/q$.

Forge. Finally, R outputs a forged signature $\sigma^* = (k_2^*, k_3^*, h^*, r^*, M^*)$ on a message m^*. After C gets σ^*, C first computes

$$\begin{aligned} K_1^* &= T_{a,1}^{r^*} M^{-(h^*+k_2^*)} \bmod p \\ K_2^* &= g^{r^*} \left(T_{s,1}^{F_1(m^*)} T_{s,2}^{F_2(m^*,g)}\right)^{-(h^*+k_2^*)} \bmod p \\ K_3^* &= g^{k_2^*} T_{v,1}^{k_3^*} \bmod p. \end{aligned} \tag{10}$$

Then, C searches $(K_1^*, K_2^*, K_3^*, M^*, m^*, h^*)$ in T_3. Because σ^* is a valid signature, R must query F_3 on $(K_1^*, K_2^*, K_3^*, M^*, M_1^*, m^*, h^*)$ previously. Thus, C can get $M_1^* = T_{v,1}^{t_{s,1}} \bmod p = g^{y\lambda_3 x\lambda_1} \bmod p$. So, $g^{xy} \bmod p = M_1^{*(\lambda_1\lambda_3)^{-1}} \bmod p$. The probability which $\sigma^* = (k_2^*, k_3^*, h^*, r^*, M^*)$ is a valid signature and did not make a F_3 query previously is $1/q$. □

Theorem 2. *The proposed SDVSUP-1 scheme is computationally nontransferable.*

Proof. The designated verifier V can simulate a valid signature σ' on the message m by the following SimSDV algorithm. V chooses randomly $x_1, x_2, r \in Z_q^*$, and $M \in Z_p^*$. Then compute

$$\begin{aligned} K_1 &= T_{a,1}^{r} M^{-x_1} \bmod p \\ K_2 &= g^{r} \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{-x_1} \bmod p \\ K_3 &= g^{x_2} \bmod p \\ M_1 &= T_{s,1}^{t_{v,1}} \bmod p. \end{aligned}$$

$$h = F_3\left(K_1, K_2, K_3, M, M_1, m\right)$$
$$k_2 = x_1 - h \bmod q$$
$$k_3 = \left(x_2 - k_2\right) t_{v,1}^{-1} \bmod q. \tag{11}$$

The simulating signature of V is $\sigma' = (k_2, k_3, h, r, M)$. Since we need the private key of S or V to verify $T_{s,1}^{t_{v,1}}$ and need the private key of S or A to verify M, anyone cannot distinguish the original signature σ and the simulating signature σ' without knowing the private keys of S, V, and A. □

Theorem 3. *The proposed SDVSUP-1 scheme is undeniable.*

Proof. When the signer S and the designated verifier V dispute who generates the signature σ on the message m, S or V submits the signature $\sigma = (k_2, k_3, h, r, M)$ on m to the arbiter A. Then, A runs the following ArbSDV algorithm. Namely, compute

$$M' = T_{s,1}^{t_{a,1}F_1(m)} T_{s,2}^{t_{a,1}F_2(m,g)} \bmod p. \tag{12}$$

Then, A checks if $M = M'$. If it is true, then A confirms that the signature σ on the message m is generated by the signer S. Otherwise, the signature σ is generated by the designated verifier V. Since M is a random number in simulating signature ϵ' while $M = T_{a,1}^{F_1(m)t_{s,1}+F_2(m,g)t_{s,2}} \bmod p$ in the real signature ϵ. Therefore, the arbiter A can use the ArbSDV algorithm to tell the real signer. □

3.4. The Proposed SDVSUP-2 Scheme. The above SDVSUP-1 scheme is a strong designated verifier signature scheme which has the undeniable property. In the SDVSUP-1 scheme, the arbiter A judges the signature generator by checking the format of M in the signature $\sigma = (k_2, k_3, h, r, M)$ on m because only M from the signer S has the format $T_{a,1}^{F(m)t_{s,1}+F(m\|g)t_{s,2}}$, while M from the designated verifier V is a random number in Z_p^*.

Because of this fact "only M from the signer S has the special format, namely, $T_{a,1}^{F(m)t_{s,1}+F(m\|g)t_{s,2}}$, while M from the designated verifier V is a random number in Z_p^*." Thus, the arbiter A only can check the format of M with the public key $(T_{s,1}, T_{s,2})$ of the signer to judge the result, which is a little unfair to the designated verifier V. In other words, the designed verifier can do nothing and it even has some doubts on the judge result. Therefore, in this subsection, we present another scheme where S and V can both construct M with their own characteristic respectively. Namely, in our SDVSUP-2 scheme, M generated by the signer S is the format $M = T_{a,1}^{F_1(m)t_{s,1}+F_2(m,g)t_{s,2}}$, while M generated by the designated verifier V is the format $M = T_{a,1}^{F_1(m)t_{v,1}+F_2(m,g)t_{v,2}}$. Thus, the arbiter A can check the format of M with the public key $(T_{s,1}, T_{s,2})$ of the signer (by computing $M = (T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)})^{t_{a,1}}$) or with the public key $(T_{v,1}, T_{v,2})$ of the designated verifier (by computing $M = (T_{v,1}^{F_1(m)} T_{v,2}^{F_2(m,g)})^{t_{a,1}}$) to judge the result, which make the arbiter A distinguish the signature easier, fairer, and more convenient. Next, we show the construction of our SDVSUP-2 scheme which is a modification of SDVSUP-1 scheme.

SetSDV. The algorithm works as the above SDVSUP-1 scheme except F_3: $(Z_q^*)^4 \times \{0,1\}^* \to Z_q^*$. Finally, the system parameters are $L = (p, q, g, F_1, F_2, F_3)$.

KeySDV. The algorithm works as the SDVSUP-1 scheme. Finally, S, V, and A obtain their private keys and public keys $((t_{s,1}, t_{s,2}), (T_{s,1}, T_{s,2}))$, $((t_{v,1}, t_{v,2}), (T_{v,1}, T_{v,2}))$, $((t_{a,1}, t_{a,2})$, and $(T_{a,1}, T_{a,2}))$, respectively.

SigSDV. S chooses randomly t_2, $r_2 \in Z_q^*$ and computes

$$K_1 = \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{t_2} \left(T_{v,1}^{F_1(m)} T_{v,2}^{F_2(m,g)}\right)^{r_2} \bmod p$$
$$K_2 = T_{a,1}^{(F_1(m)t_{s,1}+F_2(m,g)t_{s,2})(r_2+t_2)} \bmod p$$
$$M = T_{a,1}^{F_1(m)t_{s,1}+F_2(m,g)t_{s,2}} \bmod p$$
$$M_1 = T_{v,1}^{t_{s,1}} \bmod p$$
$$h = F_3\left(K_1, K_2, M, M_1, m\right)$$
$$r_1 = t_2 + h\left(F_1(m)\, t_{s,1} + F_2(m,g)\, t_{s,2}\right)^{-1} \bmod q. \tag{13}$$

The final signature on the message m is $\sigma = (r_1, r_2, h, M)$.

VerSDV. For a signature $\sigma = (r_1, r_2, h, M)$ on the message m, V computes

$$K_1' = g^{-h} \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{r_1} \left(T_{v,1}^{F_1(m)} T_{v,2}^{F_2(m,g)}\right)^{r_2} \bmod p$$
$$K_2' = T_{a,1}^{-h} M^{r_1+r_2} \bmod p$$
$$M_1 = T_{s,1}^{t_{v,1}} \bmod p$$
$$h' = F_3\left(K_1', K_2', M, M_1, m\right). \tag{14}$$

If $h = h'$, then V accepts the signature or rejects it.

Note that if we drop the inputting M_1 in the $F_3()$ of the above SDVSUP-2 scheme, namely, $h = F_3(K_1, K_2, M, m)$, then the SDVSUP-2 scheme can become a designated verifier signature not a strong scheme (namely, designated verifier signature with undeniability property (DVSUP), we call it DVSUP-2 scheme) because anyone can check the validity of the signature σ generated by DVSUP-2 scheme. Similarly, the SDVSUP-1 scheme also can become a designated verifier scheme (we call it DVSUP-1 scheme) by dropping the M_1, namely, $h = F_3(K_1, K_2, K_3, M, m)$.

3.5. Correctness of SDVSUP-2 Scheme. The above signature σ generated by S of SDVSUP-2 scheme is correct because

$$g^{-h} \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{r_1} \left(T_{v,1}^{F_1(m)} T_{v,2}^{F_2(m,g)}\right)^{r_2} \bmod p$$
$$= g^{-h} \left(T_{s,1}^{F_1(m)} T_{s,2}^{F_2(m,g)}\right)^{t_2 + h(F_1(m)t_{s,1}+F_2(m,g)t_{s,2})^{-1}} \cdot \left(T_{v,1}^{F_1(m)} T_{v,2}^{F_2(m,g)}\right)^{r_2} \bmod p$$

$$
\begin{aligned}
&= g^{-h}\left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{t_2} g^{h}\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{r_2} \bmod p \\
&= \left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{t_2}\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{r_2} \bmod p.
\end{aligned}
$$

$$
\begin{aligned}
&T_{a,1}^{-h}M^{r_1+r_2} \bmod p \\
&= T_{a,1}^{-h}M^{t_2+h(F_1(m)t_{s,1}+F_2(m,g)t_{s,2})^{-1}+r_2} \bmod p \\
&= T_{a,1}^{-h}\left(T_{a,1}^{F_1(m)t_{s,1}+F_2(m,g)t_{s,2}}\right)^{t_2+r_2} \\
&\quad\cdot\left(T_{a,1}^{F_1(m)t_{s,1}+F_2(m,g)t_{s,2}}\right)^{h(F_1(m)t_{s,1}+F_2(m,g)t_{s,2})^{-1}} \bmod p \\
&= T_{a,1}^{-h}\left(T_{a,1}^{F_1(m)t_{s,1}+F_2(m,g)t_{s,2}}\right)^{t_2+r_2} T_{a,1}^{h} \bmod p \\
&= T_{a,1}^{(F_1(m)t_{s,1}+F_2(m,g)t_{s,2})(t_2+r_2)} \bmod p.
\end{aligned} \tag{15}
$$

The above signature σ simulated by V of SDVSUP-2 scheme is correct because

$$
\begin{aligned}
&g^{-h}\left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{r_1}\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{r_2} \bmod p \\
&= g^{-h}\left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{r_1} \\
&\quad\cdot\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{t_2+h(F_1(m)t_{v,1}+F_2(m,g)t_{v,2})^{-1}} \bmod p \\
&= g^{-h}\left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{r_1}\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{t_2} \\
&\quad\cdot\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{h(F_1(m)t_{v,1}+F_2(m,g)t_{v,2})^{-1}} \bmod p \\
&= g^{-h}\left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{r_1}\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{t_2} g^{h} \bmod p \\
&= \left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{r_1}\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{t_2} \bmod p.
\end{aligned} \tag{16}
$$

$$
\begin{aligned}
&T_{a,1}^{-h}M^{r_1+r_2} \bmod p \\
&= T_{a,1}^{-h}M^{r_1+t_2+h(F_1(m)t_{v,1}+F_2(m,g)t_{v,2})^{-1}} \bmod p \\
&= T_{a,1}^{-h}\left(T_{a,1}^{F_1(m)t_{v,1}+F_2(m,g)t_{v,2}}\right)^{r_1+t_2} \\
&\quad\cdot\left(T_{a,1}^{F_1(m)t_{v,1}+F_2(m,g)t_{v,2}}\right)^{h(F_1(m)t_{v,1}+F_2(m,g)t_{v,2})^{-1}} \bmod p \\
&= T_{a,1}^{-h}\left(T_{a,1}^{F_1(m)t_{v,1}+F_2(m,g)t_{v,2}}\right)^{r_1+t_2} T_{a,1}^{h} \bmod p \\
&= T_{a,1}^{(F_1(m)t_{v,1}+F_2(m,g)t_{v,2})(r_1+t_2)} \bmod p.
\end{aligned}
$$

Theorem 4. *The proposed SDVSUP-2 scheme is computationally nontransferable.*

Proof. In order to simulate a valid signature on the message m, the designated verifier V chooses randomly t_2, $r_1 \in Z_q^*$ and computes

$$
\begin{aligned}
K_1 &= \left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{r_1}\left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{t_2} \bmod p \\
K_2 &= T_{a,1}^{(F_1(m)t_{v,1}+F_2(m,g)t_{v,2})(r_1+t_2)} \bmod p \\
M &= T_{a,1}^{F_1(m)t_{v,1}+F_2(m,g)t_{v,2}} \bmod p \\
M_1 &= T_{s,1}^{t_{v,1}} \bmod p \\
h &= F_3\left(K_1, K_2, M, M_1, m\right) \\
r_2 &= t_2 + h\left(F_1\left(m\right)t_{v,1} + F_2\left(m,g\right)t_{v,2}\right)^{-1} \bmod q.
\end{aligned} \tag{17}
$$

The final simulating signature on the message m is $\sigma' = (r_1, r_2, h, M)$. Since we need the private key of S or V to verify $T_{s,1}^{t_{v,1}}$ and need the private key of S or A to verify M, anyone cannot distinguish the original signature σ and the simulating signature σ' without knowing the private keys of S, V, and A. □

Theorem 5. *The proposed SDVSUP-2 scheme is undeniable.*

Proof. The arbiter A adapts the following method to judge the signature. A first gets the public keys of the signer and the designated verifier. Then, A uses the private $t_{a,1}$ to compute

$$
\begin{aligned}
M' &= \left(T_{s,1}^{F_1(m)}T_{s,2}^{F_2(m,g)}\right)^{t_{a,1}} \bmod p \\
M'' &= \left(T_{v,1}^{F_1(m)}T_{v,2}^{F_2(m,g)}\right)^{t_{a,1}} \bmod p.
\end{aligned} \tag{18}
$$

Then, A checks if $M = M'$ or $M = M''$. If $M = M'$, then A confirms the signature σ on the message m is generated by the signer S. If $M = M''$, then the signature σ is generated by the designated verifier V. Since $M = T_{a,1}^{F_1(m)t_{v,1}+F_2(m,g)t_{v,2}} \bmod p$ in the simulating signature ϵ', $M = T_{a,1}^{F_1(m)t_{s,1}+F_2(m,g)t_{s,2}} \bmod p$ in the real signature ϵ. Therefore, the arbiter A can use the ArbSDV algorithm to tell the real signer. □

Theorem 6. *If the CDH assumption $(t_{cdh}, \epsilon_{cdh})$ holds, then the proposed SDVSUP-2 scheme is $(t_{sdv1}, q_{f_1}, q_{f_2}, q_{f_3}, q_s, q_v, \epsilon_{sdv1})$ unforgeable.*

Proof. The proof method is very similar to the Theorem 1. So, we omit it. □

3.6. Comparison. In Tables 1 and 2, we compare our schemes with other similar schemes in terms of performance and security features. "Computational cost" denotes the totally computational cost of signing and verifying. "Signature length" denotes the signature size. "Unforg." denotes if the scheme satisfies the unforgeability property. "Nontransf." denotes if the scheme holds the nontransferability property. "Unden." denotes if the scheme holds the undeniability property. "Help from signer" denotes if it needs the help from the signer when the arbiter judges a signature's generator. "E" denotes one exponentiation computation in Z_p^*. "G_E" denotes one exponentiation computation in G where G is a bilinear group. "P" denotes one paring computation in G. "$|Z_q|$", "$|Z_p|$," and "$|G|$" denote the length of one element from "Z_q", "Z_p," and "G," respectively.

From Table 2, it can be seen that our schemes including SDVSUP-1 and SDVSUP-2 not only hold the features of

Table 1: Performance comparison with other schemes.

Scheme	Computational cost	Signature length
Jakobsson et al. [8]	$11E$	$3\|Z_q\| + 3\|Z_p\|$
Yang et al. [10]	$9E$	$4\|Z_q\|$
Tian et al. [11]	$11G_E + 2P$	$1\|Z_q\| + 4\|G\|$
Islam and Biswas [12]	$6G_E + 4P$	$1\|Z_q\| + 2\|G\|$
SDVSUP-1	$14E$	$4\|Z_q\| + 1\|Z_p\|$
SDVSUP-2	$15E$	$3\|Z_q\| + 1\|Z_p\|$

Table 2: Security features comparison with other schemes.

Scheme	Unforg.	Non-transf.	Unden.	Help from signer
Jakobsson et al. [8]	√	√	×	√
Yang et al. [10]	√	×	√	√
Tian et al. [11]	√	√	×	×
Islam and Biswas [12]	√	√	×	×
SDVSUP-1	√	√	√	×
SDVSUP-2	√	√	√	×

unforgeability and nontransferability but also have the undeniability property. What is more, the arbiter can alone judge the generator in our two schemes, while any other schemes do not have the property. Therefore, our two schemes have better security features. In terms of signature size, from Table 1, it can be seen that our schemes outperform the schemes in [8, 11] and also are comparable with the schemes in [10, 12]. In terms of computational cost, our schemes can perform many recomputations on some operations such as $T_{s,1}^{t_{v,1}}$. Therefore, our schemes also have comparable computational complexity as other schemes [8, 10–12].

3.7. Applications. SDVS has many applications such as electronic voting (e-voting) system, bidding system in business, and electronic will. Next, we demonstrate an example on how to apply our SDVSUP scheme to the bidding system in business.

We assume that B_i is a bidder on behalf of a company to make a project compete with other bidders, V is the tenderee on behalf of the enterprise to choose the most suitable company for performing the project. J is a trust third party to perform judging. Then, the bidding system consists of the following components.

Initialization Phase. Generate the system parameter L according to the SetSDV. B_i, V, and J obtain the public/private keys $((T_{B_i,1}, T_{B_i,2}), (t_{B_i,1}, t_{B_i,2}))$, $((T_{V,1}, T_{V,2}), (t_{V,1}, t_{V,2}))$, $((T_{J,1}, T_{J,2})$, and $(t_{J,1}, t_{J,2}))$, respectively, according to the KeySDV. V publishes the notice on the project for bidding.

Bidding Phase. The company B_i who wants to perform the project prepares the biding document m_i and signs on m_i with the private $(t_{B_i,1}, t_{B_i,2})$ to obtain the signature σ_i according to the SigSDV. Then B_i sends σ_i to V.

Choosing Phase. V verifies the validity of σ_i for all receiving biding documents. Then V chooses the most suitable bidder as the winner according to the price or other reasons presented in biding document.

Note. In order to obtain lowest price, V maybe show the bidding document of B_i to B_j. Thus, B_j must set lower price than B_i to obtain the project. By the similar method, V can show the bidding document of B_j to B_k who is forced to set lower price than B_j and so on, which causes a vicious cycle. An SDVS scheme can solve this problem since the V also can generate a valid signature on m_i which is indistinguishable from the original signature generated by B_i. However, if B_i and V dispute the signature σ_i on m_i or both deny the signature, then the ordinary SDVS scheme cannot solve the judge problem. But there is no such problem in our SDVSUP schemes.

Judging Phase. Given a signature σ_i on m_i, the judger J determines the signature by computing $M' = (T_{B_i,1}^{F_1(m_i)} T_{B_i,2}^{F_2(m_i,g)})^{t_{J,1}} \bmod p$ and $M'' = (T_{V,1}^{F_1(m_i)} T_{V,2}^{F_2(m_i,g)})^{t_{J,1}} \bmod p$. If $M = M'$, then σ_i is generated by the B_i. If $M = M''$, then σ_i is generated by the V.

Using the similar method to the above, our SDVSUP can be applied in electronic voting (e-voting) system, electronic will, and so on.

4. Conclusion

In this paper, we propose two strong designated verifier signature schemes including SDVSUP-1 and SDVSUP-2. Our two SDVS schemes achieve the unforgeability property, the undeniability property, and the nontransferability property. Specially, our SDVS schemes can solve the dispute of the signature ownership between the signer and the designated verifier by introducing a third party as the arbiter. The whole procedure of judgment removes the dependence on the signer and can be completed by the arbiter alone. We also present an instance on how to apply our SDVS schemes in a real situation.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by the Innovation Program of Shanghai Municipal Education Commission (no. 14ZZ167), the National Natural Science Foundation of China (nos. 61103213, 61272036, and 61672022), the Guangxi Natural Science Foundation (no. 2014GXNSFAA11838-2), and the Key Disciplines of Computer Science and Technology of Shanghai Polytechnic University (no. XXKZD1604).

References

[1] D. Chaum and H. Antwerpen, "Undeniable signatures," in *Proceedings of the 9th Annual International Cryptology Conference,*

Advances in Cryptology (CRYPTO '89), pp. 212–216, Springer, Santa Barbara, Calif, USA, August 1989.

[2] D. Chaum, "Zero-knowledge undeniable signatures (extended abstract)," in *Workshop on the Theory and Application of of Cryptographic Techniques EUROCRYPT 1990: Advances in Cryptology—EUROCRYPT '90*, vol. 473 of *Lecture Notes in Computer Science*, pp. 458–464, Springer, Berlin, Germany, 1991.

[3] R. Gennaro, T. Rabin, and R. Impagliazzo, "RSA-based undeniable signatures," *Journal of Cryptology*, vol. 13, no. 4, pp. 357–384, 2000.

[4] S. S. Duan, "Certificateless undeniable signature scheme," *Information Sciences*, vol. 178, no. 3, pp. 742–755, 2008.

[5] G. Bleumer, "Undeniable signatures," in *Encyclopedia of Cryptography and Security*, pp. 1347–1348, Springer, Berlin, Germany, 2011.

[6] M. Srinath and V. Chandrasekaran, "Isogeny-based quantum-resistant undeniable blind signature scheme," Cryptology ePrint Archive: Report 2016/148, 2016.

[7] W. Ogata, K. Kurosawa, and S.-H. Heng, "The security of the FDH variant of Chaum's undeniable signature scheme," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2006–2017, 2006.

[8] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," in *Advances in Cryptology—EUROCRYPT '96*, vol. 1070 of *Lecture Notes in Computer Science*, pp. 143–154, Springer, Berlin, Heidelberg, 1996.

[9] B. Yang, Y. Sun, Y. Yu, and Q. Xia, "A strong designated verifier signature scheme with secure disavowability," in *Proceedings of the 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS '12)*, pp. 286–291, IEEE, Bucharest, Romania, September 2012.

[10] B. Yang, Y. Yu, and Y. Sun, "A novel construction of SDVS with secure disavowability," *Cluster Computing*, vol. 16, no. 4, pp. 807–815, 2013.

[11] H. Tian, Z. Jiang, Y. Liu, and B. Wei, "A systematic method to design strong designated verifier signature without random oracles," *Cluster Computing*, vol. 16, no. 4, pp. 817–827, 2013.

[12] S. H. Islam and G. P. Biswas, "Provably secure and pairing-based strong designated verifier signature scheme with message recovery," *Arabian Journal for Science and Engineering*, vol. 40, no. 4, pp. 1069–1080, 2015.

[13] J. Ki, J. Y. Hwang, D. Nyang, B.-H. Chang, D. H. Lee, and J.-I. Lim, "Constructing strong identity-based designated verifier signatures with self-unverifiability," *ETRI Journal*, vol. 34, no. 2, pp. 235–244, 2012.

[14] H.-Y. Lin, T.-S. Wu, and S.-K. Huang, "An efficient strong designated verifier proxy signature scheme for electronic commerce," *Journal of Information Science and Engineering*, vol. 28, no. 4, pp. 771–785, 2012.

[15] Y. Ming, Q. Jin, and X. Zhao, "Designated verifier proxy signature scheme with multi-warrant in the standard model," *Journal of Information & Computational Science*, vol. 10, no. 7, pp. 2097–2107, 2013.

[16] S. H. Islam and G. Biswas, "A provably secure identity-based strong designated verifier proxy signature scheme from bilinear pairings," *Journal of King Saud University—Computer and Information Sciences*, vol. 26, no. 1, pp. 55–67, 2014.

[17] J. Wang, Q. Guo, and Y. Wang, "Security analysisi of a designated-verifier proxy signature scheme," *Journal of Northwest Normal University (Natural Science)*, vol. 51, no. 5, pp. 55–58, 2015.

[18] H. Lipmaa, G. Wang, and F. Bao, "Designated verifier signature schemes: attacks, new security notions and a new construction," in *Automata, Languages and Programming*, vol. 3580 of *Lecture Notes in Computer Science*, pp. 459–471, Springer, Berlin, Germany, 2005.

[19] 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.

[20] Q. Huang, G. Yang, D. S. Wang, and W. Susilo, "Efficient strong designated verifier signature schemes without random oracle or with non-delegability," *International Journal of Information Security*, vol. 10, no. 6, pp. 373–385, 2011.

[21] H.-Y. Lin, T.-S. Wu, and S.-K. Huang, "An efficient strong designated verifier proxy signature scheme for electronic commerce," *JISE. Journal of Information Science and Engineering*, vol. 28, no. 4, pp. 771–785, 2012.

[22] R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk, "Universal designated verifier signatures," in *International Conference on the Theory and Application of Cryptology and Information Security ASIACRYPT 2003: Advances in Cryptology—ASIACRYPT 2003*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 523–542, Springer, Berlin, Germany, 2003.

[23] X. Huang, W. Susilo, Y. Mu, and W. Wu, "Secure universal designated verifier signature without random oracles," *International Journal of Information Security*, vol. 7, no. 3, pp. 171–183, 2008.

[24] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "Short designated verifier signature scheme and its identity-based variant," *International Journal of Network Security*, vol. 6, no. 1, pp. 82–93, 2008.

[25] H. Krawczyk and T. Rabin, "Chameleon hashing and signatures," in *Proceedings of the Network and Distributed System Security Symposium*, pp. 143–154, San Diego, Calif, USA, 2000.