Certificateless multi-signer universal designated multi-verifier signature from elliptic curve group

Lunzhi Deng[1,2,]*, Yixian Yang[2], Yuling Chen[2]
[1]School of Mathematical Sciences, Guizhou Normal University, Guiyang 550001, China
[2]Guizhou University, Guizhou Provincial Key Laboratory of Public Big Data, Guiyang, 550025, China
[e-mail: denglunzhi@163.com]

Received April 6, 2016; revised April 11, 2017; revised May 27, 2017; accepted July 7, 2017; published November 30, 2017

Abstract

Certificateless public key cryptography resolves the certificate management problem in traditional public key cryptography and the key escrow problem in identity-based cryptography. In recent years, some good results have been achieved in speeding up the computation of bilinear pairing. However, the computation cost of the pairing is much higher than that of the scalar multiplication over the elliptic curve group. Therefore, it is still significant to design cryptosystem without pairing operations. A multi-signer universal designated multi-verifier signature scheme allows a set of signers to cooperatively generate a public verifiable signature, the signature holder then can propose a new signature such that only the designated set of verifiers can verify it. Multi-signer universal designated multi-verifier signatures are suitable in many different practical applications such as electronic tenders, electronic voting and electronic auctions. In this paper, we propose a certificateless multi-signer universal designated multi-verifier signature scheme and prove the security in the random oracle model. Our scheme does not use pairing operation. To the best of our knowledge, our scheme is the first certificateless multi-signer universal designated multi-verifier signature scheme.

Keywords: Certificatelss cryptography, Multi-user setting, Universal designated verifier signature.

https://doi.org/10.3837/tiis.2017.11.025 **ISSN : 1976-7277**

1. Introduction

Traditional public key infrastructure requires a trusted certification authority to issue a certificate binding the identity to the public key of an entity. So we are confronted with the problem of certificate management. To solve the problem, Shamir [1] defined identity based public key cryptography, which needs a trusted private key generator (PKG) to generate a private key for an entity according to his identity. However, it brings the key escrow problem. To solve the two problems, Al-Riyami et al. [2] introduced certificateless public key cryptography, which requires a semi-trusted key generation center (KGC) to generate user's partial private key with respect to user's identity. A user's full private key includes two parts: partial private key generated by KGC and a secret value chosen by himself.

In some practical applications such that electronic voting, electronic tenders and electronic auctions, the public verification and non-repudiation properties of a signature are not desired. For an example, in electronic voting schemes, a voting center aim to verify that a vote has been properly counted in the final tally by means of the center’s signature on the receipt. Voters must not have the capability to use such receipts to verify the nature of their votes. Otherwise, it results that the briber is sure that the voter indeed votes some candidate according to his/her direction and then give the obedient voter some gain.

In 1996, Jakobsson et al. [3] introduced the designated verifier signature concept (DVS). In this setting, only the designated verifier is able to verify the validity of the signature. Other anyone can not be convinced of the authenticity of this signature since the designated verifier himself can simulate the signature which is indistinguishable from the one generated by the signer. In 2003, Saeednia et al. [4] formalized strong designated verifier signature. In this notion, the designated verifier’s private key is involved in the verification phase so that other anyone cannot verify the validity of the signature. In 2003, Steinfeld et al. [5] defined universal designated verifier signature. It has the same properties as DVS schemes and it also can function as a standard publicly verifiable signature scheme that additionally allows any signature holder (not necessarily the signer) to design the signature to any desired designated verifier.

1.1. Related work

Laguillaumie and Vergnaud [6, 7] developed designated multi-verifier signature , in which the signer can designate a set of verifier. Ng et al. [8] introduced universal designated multi-verifier signature, which allow a signature holder to designate the signature to multi-verifier. Zhang et al. [9] proposed a multi-signer strong designated verifier signature. In this setting, a set of signers collaborate to generate a signature such that only the designated user can verify it with his own private key.

Shailaja et al.[10] and Ming et al. [11] proposed a universal designated verifier signature scheme, respectively. The schemes[10,11] were proved to be secure in standard model. Roeder et al.[12] proposed practical constructions of multi-verifier signature schemes that are provably secure and are based only on pseudorandom functions in the plain model. Zhang et al.[13] proposed a strong multi-designated verifiers signature, which is secure against rogue key attack. Zhang et al. [14] proposed the model of ID-based universal designated verifier signature scheme and provided two concrete constructions. Kang et al. [15] proposed the model of ID-based strong designated verifier signature scheme and provided a concrete construction, the size of signature is only two elements. Lin [16] proposed an ID-based toward secure strong designated verifier signature scheme, which can resist the key-compromise attack. Seo et al. [17] proposed an ID-based universal designated multi-verifier signature scheme with constant signature size. Chang [18] proposed an ID-based multi-signer universal designated multi-verifier signature, which allows a set of multi-signer to cooperatively generate a signature and designate a set of multi-verifier to verify it. Ming et al. [19] proposed a multi-signer universal designated multi-verifier signature scheme in the standard model.

Huang et al.[20] proposed the first certificateless designated verifier signature scheme. Ming et al.[21] constructed the first certificateless universal designated verifier signature scheme. The security of two schemes [20,21] relies on Bilinear Diffie-Hellman Problem. Yang et al. [22] and Du et al [23] proposed an efficient certificateless designated verifier signature scheme, respectively, the two schemes [22, 23] use the bilinear pairing. He and Chen [24] proposed another efficient certificateless designated verifier signature scheme based on elliptic curve group without bilinear pairing.

1.2. Motivations and our contributions

Some good results have been achieved in speeding up the computation of pairing function in recent years. However, the computation cost of the pairing is much higher than that of the scalar multiplication over the elliptic curve group. Therefore, it is quite significant to design cryptosystem without pairing operation. Although designated multi-verifier signatures have been studied extensively over the last few years, all schemes currently known were constructed from ID-based setting. There is not any work published from certificateless setting.

In this paper, we propose the first certificateless multi-signer universal designated multi-verifier signature (CS^{M}UDV^{M}S) scheme which have the following features:

(1) The scheme is proven to be secure under the random oracle model.
(2) The scheme does not use pairing operation and it is more efficient than previous ones.

2. Definitions

Definition 1. Let E/F_p be an elliptic curve, $P \in E$ is a point having prime order q. Let $G = (P) \le E$, given $aP, bP \in G$, the computational Diffie-Hellman problem (CDHP) is to compute abP.

2.1. System model

A CS^{M}UDV^{M}S scheme consists of the following nine algorithms:

Setup: On input a security parameter v, this algorithm outputs the system parameters $params$ and the master secret key msk. It is run by PKG

Partial private key extract: On input a user's identity $ID_i \in \{0,1\}^*$, this algorithm outputs the partial private key D_i. It is run by PKG.

Secret value set: On input a user's identity $ID_i \in \{0,1\}^*$, this algorithm outputs the secret value u_i. It is run by user himself.

User public key generate: On input a user's identity $ID_i \in \{0,1\}^*$, this algorithm outputs the public key $Q_i =（P_i, X_i）$, where X_i is generated by KGC, P_i is generated by user himself. It is run by user himself.

C-S: On input a set $\aleph = W \bigcup \{Q_{Si} : ID_{Si} \in W\}$, where W is the set of signers' identities, this algorithm outputs n individual signatures σ_i on a message M , It is run by users themselves.

C-SV: On input a signature σ_i on message M under the identity ID_{Si}, this algorithm outputs 1 if the signature is valid or 0 otherwise. It is run by any verifier.

C-S^{M}: On input n individual signatures $\sigma_i (i = 1,2,\cdots,n)$, this algorithm outputs a multi-signer signature $\overline{\sigma}$ on message M . It is run by any signer.

C-S^{M}V: On input a multi-signer signature $\overline{\sigma}$ on message M under the signer set W, this algorithm outputs 1 if the signature is valid or 0 otherwise. It is run by any verifier.

C-SMUDVMS: On input a message-signature pair （$M,\overline{\sigma}$) and a set $\Re = V \bigcup \{Q_{Vj} : ID_{Vj} \in V\}$, where V is the set of verifiers' identities, this algorithm outputs a certificateless multi-signer universal designated multi-verifier signature δ on message M . It is run by anyone hold the tuple（$M,\overline{\sigma}$) .

C-S^{M}UDV^{M}SV: On input a multi-signer universal designated multi-verifier signature δ on message M , this algorithm outputs 1 if the signature is valid or 0 otherwise. It is run by the designated verifiers.

2.2. Security model

Definition 2. A certificateless multi-signer universal designated multi-verifier signature scheme is unforgeable (UNF-CS^{M}UDV^{M}S) if the advantage of any polynomial bounded adversary is negligible in the following two games against Type I/II adversaries.

Game I. Now we illustrate the game performed between a challenger C and a Type I adversary A_I for a CS^{M}UDV^{M}S scheme.

Initialization. The challenger C runs the setup algorithm with a security parameter v and gives the system parameters to the adversary A_I .

Query. The adversary A_I performs a polynomial bounded number of queries.

User public key query: A_I can request the user public key for any user.

User public key replacement request: A_I supplies a new value $P_i^{/}$ with respect to a user ID_i, C replaces the current public key P_i with $P_i^{/}$. Note that A_I can replace only the part of public key generated by user himself.

Hash functions query: A_I can ask for the values of the hash functions for any input.

Partial private key query: A_I requests the partial private key of a user ID_i, C responds with the partial private key D_i .

Secret value query: A_I requests the secret value of a user ID_i . C returns the secret value u_i, if a user's public key was replaced, A_I cannot request the corresponding secret value.

Signature query: A_I gives a signer set W , a receiver set V and a message M , C then gives a signature δ on the message M .

Forge. A_I outputs a new tuple （$\delta, M, \aleph, \Re$). The adversary wins if the result of Verify （$\delta, M, \aleph, \Re$) is the symbol 1 and the following conditions hold:

(1) A_I cannot query the partial private key of anyone in W .

(2) The forged signature （$\delta, M, \aleph, \Re$) is not from signature query.

The advantage of A_I is defined as: Adv $A_I^{UNF-CS^M UDV^M S}$ = P_r [A_I wins].

Game II. Now we illustrate the game performed between a challenger C and a Type II adversary A_{II} for a CS^{M}UDV^{M}S scheme.

Initialization. The challenger C runs the setup algorithm with a security parameter v and gives the system parameters and master secret key to the adversary A_{II} .

Query. The adversary A_{II} performs a polynomial bounded number of queries as those in Game I

Forge. A_{II} outputs a new tuple （$\delta, M, \aleph, \Re$). The adversary wins if the result of Verify （$\delta, M, \aleph, \Re$) is the symbol 1 and the following conditions hold:

(1) A_{II} can not query the secret value of anyone in W .

(2) A_{II} can not replace the public key of anyone in W .

(3) The forged signature（$\delta, M, \aleph, \Re$) is not from signature query.

The advantage of A_{II} is defined as: Adv $A_{II}^{UNF-CS^M UDV^M S}$ = P_r [A_{II} wins].

Remark. For a forge signature （$\delta, M, \aleph, \Re$)**,** A_I can know the secret value of anyone in W , however, he does not know the partial private key of anyone in W**.** On the other hand, A_{II} can know the partial private key of anyone in W , however, he does not know the secret value of anyone in W**.**

Definition 3. A certificateless multi-signer universal designated multi-verifier signature scheme is non-transferability if each designated verifier cannot convince anyone of the authenticity of the signature δ on the message M , even if all private keys of signers are revealed.

3. Our scheme

Setup: Given the security parameter v , KGC chooses an elliptic curve group G of prime order $q > 2^v$, a generator P of G . Then, KGC chooses three cryptographic hash functions $H_1, H_2, H_3 : \{0,1\}^* \to Z_q^*$, Finally, PKG chooses master secret key $\lambda \in Z_q^*$ and sets the public key $P_{pub} = \lambda P$. The set of public parameters is: $params = \{G, q, P_{pub}, H_1, H_2, H_3\}$

Partial private key extract: For a user $ID_i \in \{0,1\}^*$, KGC picks $x_i \in Z_q^*$ at random and computes $X_i = x_i P$, $c_i = H_1(ID_i, X_i)$, $d_i = x_i + c_i \lambda$, then sends $D_i = (X_i, d_i)$ to the user.

Secret value set: The user ID_i randomly chooses $u_i \in Z_q^*$

User public key generate: The user ID_i computes $P_i = u_i P$, and sets $Q_i = (P_i, X_i)$ as his public key.

C-S: Let $\aleph = W \bigcup \{Q_{Si} : ID_{Si} \in W\}$, $W = \{ID_{S1}, \cdots, ID_{Sn}\}$ is the set of signers' identities, each signer performs the following steps to generate his individual signature on a message M :

(1) Randomly chooses $r_i, t_i \in Z_q^*$, computes $R_i = r_i P$, $T_i = t_i P$ and then broadcasts (R_i, T_i) to other co-signers.

(2) Computes $\overline{R} = \sum_{i=1}^{n} R_i$, $\overline{T} = \sum_{i=1}^{n} T_i$, $h = H_2(M, \aleph, \overline{R}, \overline{T})$, $k = H_3(M, \aleph, \overline{R}, \overline{T})$, $y_i = r_i + hd_{Si}$ and $z_i = t_i + ku_{Si}$.

The certificateless individual signature on message M is $\sigma_i = (y_i, z_i, R_i, \overline{R}, T_i, \overline{T})$.

C-SV: A verifier can check whether a signature $\sigma_i = (y_i, z_i, R_i, \overline{R}, T_i, \overline{T})$ on message M is given by ID_{Si} as follows:

(1) Computes $c_{Si} = H_1(ID_{Si}, X_{Si})$, $h = H_2(M, \aleph, \overline{R}, \overline{T})$, $k = H_3(M, \aleph, \overline{R}, \overline{T})$.

(2) Checks if $y_i P = R_i + h(X_{Si} + c_{Si} P_{pub})$, $z_i P = T_i + kP_{Si}$. If the equalities hold, outputs 1. Otherwise, outputs 0.

C-S$^{\mathbf{M}}$: Combines n co-signers' individual signature $\sigma_i (i = 1,2,\cdots,n)$: $\overline{y} = \sum_{i=1}^{n} y_i$. $\overline{z} = \sum_{i=1}^{n} z_i$. Outputs the signature $\overline{\sigma} = (\overline{y}, \overline{z}, \overline{R}, \overline{T})$ on message M .

C-S$^{\mathbf{M}}$V: A verifier can check whether a signature $\overline{\sigma} = (\overline{y}, \overline{z}, \overline{R}, \overline{T})$ on message M is given by n signers W as follows:

(1) Computes $h = H_2(M, \aleph, \overline{R}, \overline{T})$, $k = H_3(M, \aleph, \overline{R}, \overline{T})$, $c_{Si} = H_1(ID_{Si}, X_{Si})$ for $i = 1,2,\cdots,n$.

(2) Checks if $\overline{y}P = \overline{R} + h\sum_{i=1}^{n}(X_{Si} + c_{Si}P_{pub})$, $\overline{z}P = \overline{T} + k\sum_{i=1}^{n} P_{Si}$. If the equalities hold, outputs 1. Otherwise, outputs 0.

C-S$^{\mathbf{M}}$UDV$^{\mathbf{M}}$S: Let $\Re = V \bigcup \{Q_{Vj} : ID_{Vj} \in V\}$, $V = \{ID_{V1}, \cdots, ID_{Vm}\}$ is the set of verifiers' identities. Given a set V and a message-signature pair $(M, \overline{\sigma})$, anyone can give an C-S^{M}UDV^{M}S as follows:

(1) Computes $c_{Vj} = H_1(ID_{Vj}, X_{Vj})$ for $j = 1,2,\cdots,m$.

(2) Computes $Y = \overline{y} \cdot \sum_{j=1}^{m}(X_{Vj} + c_{Vj}P_{pub})$, $Z = \overline{z} \cdot \sum_{j=1}^{m} P_{Vj}$

(3) Outputs the signature $\delta = (Y, Z, \overline{R}, \overline{T})$ on message M .

C-S$^{\mathbf{M}}$UDV$^{\mathbf{M}}$SV: The designated verifier set V can check whether a signature $\delta = (Y, Z, \overline{R}, \overline{T})$ on message M is given by the signer set W as follows:

(1) Computes $h = H_2(M, \aleph, \overline{R}, \overline{T})$, $k = H_3(M, \aleph, \overline{R}, \overline{T})$, $c_{Si} = H_1(ID_{Si}, X_{Si})$ for $i = 1,2,\cdots,n$.

(2) Checks if $Y = \sum_{j=1}^{m} d_{Vj} \cdot (\overline{R} + h(\sum_{i=1}^{n}(X_{Si} + c_{Si}P_{pub}))$, $Z = \sum_{j=1}^{m} u_{Vj} \cdot (\overline{T} + k \cdot \sum_{i=1}^{n} P_{Si})$ .If the equalities hold, outputs 1. Otherwise, outputs 0.

On correctness, we have

$$\sum_{j=1}^{m} d_{Vj} \cdot (\overline{R} + h(\sum_{i=1}^{n}(X_{Si} + c_{Si}P_{pub})) = \sum_{j=1}^{m}(x_{Vj} + \lambda c_{Vj}) \cdot (\sum_{i=1}^{n} r_i P + h(\sum_{i=1}^{n}(x_{Si}P + \lambda c_{Si}P))$$

$$= \sum_{j=1}^{m}(x_{Vj} + \lambda c_{Vj})P \cdot \sum_{i=1}^{n}(r_i + hd_{Si}) = \sum_{i=1}^{n} y_i \cdot \sum_{j=1}^{m}(X_{Vj} + c_{Vj}P_{pub})$$

$$= \overline{y} \cdot \sum_{j=1}^{m}(X_{Vj} + c_{Vj}P_{pub}) = Y$$

$$\sum_{j=1}^{m} u_{Vj} \cdot (\overline{T} + k \cdot \sum_{i=1}^{n} P_{Si}) = \sum_{j=1}^{m} u_{Vj} \cdot (\sum_{i=1}^{n} t_i P + k \cdot \sum_{i=1}^{n} u_{Si}P) = \sum_{j=1}^{m} u_{Vj}P \cdot \sum_{i=1}^{n}(t_i + ku_{Si})$$

$$= \sum_{j=1}^{m} u_{Vj}P \cdot \sum_{i=1}^{n} z_i = \sum_{i=1}^{n} z_i \cdot \sum_{j=1}^{m} P_{Vj} = \overline{z} \cdot \sum_{j=1}^{m} P_{Vj} = Z$$

4. Security results

Theorem 1. The scheme is unforgeable against the super Type I adversary in randomly oracle model if the CDHP is hard.

Proof. Suppose the challenger C receives a random instance (P, aP, bP) of the CDHP and has to compute abP. C will run A_I as a subroutine and act as A_I's challenger in the game.

Initialization. C runs the setup algorithm with the parameter v, then gives A_I the system parameters $params = \{G, q, P_{pub} = \lambda P, H_1, H_2, H_3\}$.

Queries. Without loss of generality, we assume that all the queries are distinct and A_I will ask for user public key before an identity ID is used in any other queries. C will set several lists to store the queries and answers, all of the lists are initially empty.

User public key queries: C maintains the list L_U of tuple (ID_i, u_i, x_i). When A_I issues a user public key query for ID_i, C responds as follows:
At the j^{th} query , randomly picks $u_j \in Z_q^*$, sets $ID_j = ID^*$ and $Q_j = (u_j P, aP)$. At the f^{th} query, randomly picks $u_f \in Z_q^*$, sets $ID_f = ID^\#$ and $Q_f = (u_f P, bP)$. For $i \neq j, f$, C randomly picks $x_i, u_i \in Z_q^*$ and returns $Q_i = (u_i P, x_i P)$, then the query and the answer will be stored in list L_U.

User public key replacement requests: C maintains the list L_R of tuple $(ID_i, P_i, P_i^/)$. When A_I issues a user public key replacement request for ID_i with a new $P_i^/$, C replaces the current public key value P_i with $P_i^/$ and adds $(ID_i, P_i, P_i^/)$ to list L_R.
H_1 *queries:* C maintains the list L_1 of tuple (α_i, c_i). When A_I issues a query $H_1(\alpha_i)$, C randomly picks $c_i \in Z_q^*$, sets $H_1(\alpha_i) = c_i$ and adds (α_i, c_i) to list L_1.

H_2 *queries:* C maintains the list L_2 of tuple (β_i, h_i). When A_I issues a query $H_2(\beta_i)$, C randomly picks $h_i \in Z_q^*$, sets $H_2(\beta_i) = h_i$ and adds (β_i, h_i) to list L_2.

H_3 *queries:* C maintains the list L_3 of tuple (γ_i, k_i).When A_I issues a query $H_3(\gamma_i)$, C randomly picks $k_i \in Z_q^*$, sets $H_3(\gamma_i) = k_i$ and adds (γ_i, k_i) to list L_3.

Partial private key queries: C maintains the list L_D of tuple (ID_i, D_i).When A_I issues a partial private key query for identity ID_i. If $ID_i = ID^*$ or $ID_i = ID^\#$, A_I fails and stops. Otherwise, C finds the tuple (ID_i, u_i, x_i) in list L_U, gives the D_i by calling the private key extract algorithm, then adds (ID_i, D_i) to list L_D.

Secret value queries: When A_I issues a secret key query for identity ID_i. C checks list L_U, finds (ID_i, u_i, x_i) in list L_U, responds with u_i.

Signature queries: When A_I supplies a set $\aleph = W \bigcup \{Q_{Si} : ID_{Si} \in W\}$ ($W = \{ID_{S1}, \cdots, ID_{Sn}\}$ is the set of signers' identities), a set $\Re = V \bigcup \{Q_{Vj} : ID_{Vj} \in V\}$ ($V = \{ID_{V1}, \cdots, ID_{Vm}\}$ is the set of verifiers' identities), and a message M, C outputs a CS^{M}UDV^{M}S as follows:

(1) Computes $c_{Si} = H_1(ID_{Si}, X_{Si})$ for $i = 1,2,\cdots,n$.

(2) Randomly chooses $h, k, y_i, z_i \in Z_q^*$ for $i = 1,2,\cdots,n$.

(3) Computes $R_i = y_i P - h(X_{Si} + c_{Si} P_{pub})$ for $i = 1,2,\cdots,n$, $\overline{R} = \sum_{i=1}^{n} R_i$.

(4) Computes $T_i = z_i P - kP_{Si}$ for $i = 1,2,\cdots,n$, $\overline{T} = \sum_{i=1}^{n} T_i$.

(5) Adds $h = H_2(M, \aleph, \overline{R}, \overline{T})$, $k = H_3(M, \aleph, \overline{R}, \overline{T})$ to list L_2 and L_3, respectively. If collision occurs, repeats the steps (2)-(5).

(6) Computes $\overline{y} = \sum_{i=1}^{n} y_i$, $\overline{z} = \sum_{i=1}^{n} z_i$, $Y = \overline{y} \cdot \sum_{j=1}^{m} (X_{Vj} + c_{Vj} P_{pub})$ and $Z = \overline{z} \cdot \sum_{j=1}^{m} P_{Vj}$.

(7) Outputs the signature $\delta = (Y, Z, \overline{R}, \overline{T})$ on message M.

Forge. A_I outputs a forged signature $\delta = (Y, Z, \overline{R}, \overline{T})$ on message M, signer set W, verifier set V and fulfills the following conditions:

1. A_I can not query the partial private key of anyone in W.
2. The forged signature $\delta = (Y, Z, \overline{R}, \overline{T})$ is not from signature query.

Solve CDHP. By the forking lemma for signature scheme [25], that if A_I can give a valid signature $\delta = (Y, Z, \overline{R}, \overline{T})$ on message M with probability ε in the above interaction, then there exists another algorithm $A_I^{/}$ that can output another valid signature $\delta^{/} = (Y^{/}, Z, \overline{R}, \overline{T})$ on message M for same signer set W and verifier set V. To do so we keep all the random tapes in two invocations of the same except h returned by H_2 query of the forged message, so we have $h \neq h^{/}$. If $ID^* \in W$, $ID^\# \in V$ or $ID^\# \in W$, $ID^* \in V$, then we can solve CDHP . Without loss of generality, we may assume that $ID_{S1} = ID^* \in W$, $ID_{V1} = ID^\# \in V$, then

$$Y = (b + c_{V1}\lambda + \sum_{j=2}^{m} d_{Vj})(\overline{R} + h(aP + \sum_{i=2}^{n} X_{Si} + \sum_{i=1}^{n} c_{Si} P_{pub})) \ ,$$

$$Y^{/} = (b + c_{V1}\lambda + \sum_{j=2}^{m} d_{Vj})(\overline{R} + h^{/}(aP + \sum_{i=2}^{n} X_{Si} + \sum_{i=1}^{n} c_{Si} P_{pub})).$$

We can get abP as follows:

(1) Find $(ID_{Vj}, u_{Vj}, x_{Vj})$ in list L_U, compute $X_{Vj} = x_{Vj}P$, $c_{Vj} = H_1(ID_{Vj}, X_{Vj})$ for $j = 2,\cdots,m$.

(2) Compute $d_{Vj} = x_{Vj} + c_{Vj}\lambda$ for $j = 2,3\cdots,m$.

(3) Find $(ID_{Si}, u_{Si}, x_{Si})$ in list L_U, compute $X_{Si} = x_{Si}P$, $c_{Si} = H_1(ID_{Si}, X_{Si})$ for $i = 1,2,\cdots,n$.

(4) Compute

$$abP = (h - h')^{-1}(Y - Y') - (c_{V1} + \sum_{j=2}^{m} d_{Vj})(aP + \sum_{i=2}^{n} X_{Si} + \sum_{i=1}^{n} c_{Si}P_{pub}) - \sum_{i=2}^{n} x_{Si}bP - \sum_{i=1}^{n} c_{Si}\lambda bP$$

Probability. Let q_U, q_D and q_S be the numbers of user public key queries, partial private key queries and signature queries, respectively.

The probability that C does not fail during the partial private key queries is

$$\frac{(q_U - q_D)(q_U - q_D - 1)}{q_U(q_U - 1)}.$$

The probability that $ID^* \in W$, $ID^\# \in V$ or $ID^\# \in W$, $ID^* \in V$ at least is

$$\frac{2mn}{q_U(q_U - q_D)}.$$

So the combined probability is

$$\frac{(q_U - q_D)(q_U - q_D - 1)}{q_U(q_U - 1)} \frac{2mn}{q_U(q_U - q_D)} > \frac{2(q_U - q_D - 1)}{q_U^2(q_U - 1)}$$

Therefore, if A_I can give a valid forge signature with probability ε within time T, then C can solve the CDHP with the probability $\frac{2(q_U - q_D - 1)}{q_U^2(q_U - 1)}\varepsilon$. The running time required for C is $2T + (2q_U + q_D + (5n + m + 2)q_S) \cdot T_S$, where T_S denotes the time for a scalar point multiplication in G.

Theorem 2. The scheme is unforgeable against the super Type II adversary in randomly oracle model if the CDHP is hard.

Proof. Suppose the challenger C receives a random instance (P, aP, bP) of the CDHP and has to compute abP. C will run A_{II} as a subroutine and act as A_{II}'s challenger in the game.

Initialization. C runs the setup algorithm with the parameter v, then gives A_{II} the system parameters $params = \{G, q, P_{pub} = \lambda P, H_1, H_2, H_3\}$ and master secret key λ

Queries. Without loss of generality, we assume that all the queries are distinct and A_{II} will ask for user public key before an identity ID is used in any other queries. C will set several lists to store the queries and answers, all of the lists are initially empty.

User public key queries: C maintains the list L_U of tuple (ID_i, u_i, x_i). When A_{II} issues a user public key query for ID_i, C responds as follows:
At the j^{th} query , randomly picks $x_j \in Z_q^*$, sets $ID_j = ID^*$ and $Q_j = (aP, x_j P)$. At the f^{th} query, randomly picks $x_f \in Z_q^*$, sets $ID_f = ID^{\#}$ and $Q_f = (bP, x_f P)$. For $i \neq j, f$, C randomly picks $x_i, u_i \in Z_q^*$ and returns $Q_i = (u_i P, x_i P)$, then the query and the answer will be stored in list L_U.

User public key replacement requests, H_1, H_2, H_3 *queries:* Same as that in Theorem 1.

Partial private key queries: C maintains the list L_D of tuple (ID_i, D_i) .When A_{II} issues a partial private key query for identity ID_i. C finds the tuple (ID_i, u_i, x_i) in list L_U, gives the D_i by calling the private key extract algorithm, then adds (ID_i, D_i) to list L_D.

Secret value queries: When A_{II} issues a secret key query for identity ID_i. If $ID_i = ID^*$ or $ID_i = ID^{\#}$, A_{II} fails and stops. Otherwise, C checks list L_U, finds (ID_i, u_i, x_i) in list L_U, responds with u_i.

Signature queries: Same as that in Theorem 1.
Forge. A_{II} outputs a forged signature $\delta = (Y, Z, \overline{R}, \overline{T})$ on message M for signer set W , verifier set V and fulfills the following conditions:
1. A_{II} can not query the secret value of anyone in W.
2. A_{II} can not replace the public key of anyone in W.
3. The forged signature $\delta = (Y, Z, \overline{R}, \overline{T})$ is not from signature query.

Solve CDHP. By the forking lemma for signature scheme [25], that if A_{II} can give a valid signature $\delta = (Y, Z, \overline{R}, \overline{T})$ with probability ε in the above interaction, then there exists another algorithm $A_{II}^{/}$ that can output another valid signature $\delta^{/} = (Y, Z^{/}, \overline{R}, \overline{T})$ on message M for same signer set W and verifier set V. To do so we keep all the random tapes in two invocations of the same except h returned by H_3 query of the forged message, so we have $k \neq k^{/}$. If $ID^* \in W$, $ID^{\#} \in V$ or $ID^{\#} \in W$, $ID^* \in V$, then we can solve CDHP . Without loss of generality, we may assume that $ID_{S1} = ID^* \in W$, $ID_{V1} = ID^{\#} \in V$, then

$$Z = (b + \sum_{j=2}^{m} u_{Vj})(\overline{T} + k(aP + \sum_{i=1}^{n} P_{Si})) \quad ,$$

$$Z^{/} = (b + \sum_{j=2}^{m} u_{Vj})(\overline{T} + k^{/}(aP + \sum_{i=1}^{n} P_{Si})) .$$

We can get abP as follows:

(1) Find $(ID_{Vj}, u_{Vj}, x_{Vj})$ in list L_U, compute $P_{Vj} = u_{Vj}P$ for $j = 2, \cdots, m$.

(2) Find $(ID_{Vj}, u_{Si}, x_{Si})$ in list L_U, compute $P_{Si} = u_{Si}P$ for $i = 1,2,\cdots,n$.

(3) Compute $abP = (k - k^{/})^{-1}(Z - Z^{/}) - \sum_{i=1}^{n} u_{Si} \cdot bP - \sum_{j=2}^{m} u_{Vj} \cdot (aP + \sum_{i=1}^{n} P_{Si})$

Probability. Let q_U, q_E and q_S be the numbers of user public key queries, secret value queries and signature queries, respectively.

The probability that C does not fail during the partial private key queries is

$$\frac{(q_U - q_E)(q_U - q_E - 1)}{q_U(q_U - 1)} .$$

The probability that $ID^* \in W$, $ID^\# \in V$ or $ID^\# \in W$, $ID^* \in V$ at least is

$$\frac{2mn}{q_U(q_U - q_E)} .$$

So the combined probability is

$$\frac{(q_U - q_E)(q_U - q_E - 1)}{q_U(q_U - 1)} \frac{2mn}{q_U(q_U - q_E)} > \frac{2(q_U - q_E - 1)}{q_U^2(q_U - 1)}$$

Therefore, if A_{II} can give a valid forge signature with probability ε within time T, then C can solve the CDHP with the probability $\frac{2(q_U - q_E - 1)}{q_U^2(q_U - 1)}\varepsilon$. The running time required for C is $2T + (2q_U + (5n + m + 2)q_S) \cdot T_S$, where T_S denotes the time for a scalar point multiplication in G.

Theorem 3. The scheme is non-transferability.

Proof. Given a CS^{M}UDV^{M}S $\delta = (Y, Z, \overline{R}, \overline{T})$ on a signer set $W = \{ID_{S1}, \cdots, ID_{Sn}\}$ and a message M, the designated multi-verifier $V = \{ID_{V1}, \cdots, ID_{Vm}\}$ can always produce a valid CS^{M}UDV^{M}S $\delta^{/} = (Y^{/}, Z^{/}, \overline{R}^{/}, \overline{T}^{/})$ on the message M as follows:

(1) Set $\aleph = W \bigcup \{Q_{Si} : ID_{Si} \in W\}$.

(2) Randomly choose $r_i^{/}, t_i^{/} \in Z_q^*$, for $i = 1,2,\cdots,n$.

(3) Compute

$$\overline{R}^{/}=\sum_{i=1}^{n} r_i^{/}P,\overline{T}^{/}=\sum_{i=1}^{n} t_i^{/}P, h=H_2(M,\aleph,\overline{R}^{/},\overline{T}^{/}), k=H_3(M,\aleph,\overline{R}^{/},\overline{T}^{/}).$$

(4) Compute

$$Y^{/}=\sum_{j=1}^{m} d_{Vj}\cdot(\overline{R}^{/}+h(\sum_{i=1}^{n}(X_{Si}+\sum_{i=1}^{n} c_{Si}P_{pub}))\ , Z^{/}=\sum_{j=1}^{m} u_{Vj}\cdot(\overline{T}^{/}+k\sum_{i=1}^{n} P_{Si})$$

It is clearly that the signature $\delta^{/}=(Y^{/},Z^{/},\overline{R}^{/},\overline{T}^{/})$ on the message M can pass algorithm CS^{M}UDV^{M}SV.

So given a message M, the distribution of $\delta^{/}=(Y^{/},Z^{/},\overline{R}^{/},\overline{T}^{/})$ is perfectly indistinguishable from that of $\delta=(Y,Z,\overline{R},\overline{T})$. It is unconditionally infeasible to determine who the original signer set W and the designated verifiers V generate this signature, even if all private keys are revealed. Thus, the proposed scheme achieves non-transferability.

5. Efficiency

In this section, we compare the performance of our scheme with other schemes. We use SL to denote signature length and denote some notations as follows.

P : a pairing operation.

E_G : a pairing-based scalar multiplication operation.

E_S : a scalar multiplication operation.

n : the number of signers.

m : the number of verifiers.

Through an Intel I7-3537U 2.50GHz processor, 8G bytes memory and the Window 7 operating system, we obtained the running time for cryptographic operations. To achieve the 1024-bit RSA level security, we used the Ate pairing $e:G_1\times G_1\rightarrow G_2$, where G_1 with order q is generated by a point on a super singular elliptic curve $E/F_p: y^2=x^3+x$ defined on the finite field F_p, q is a 160-bits prime number, and p is a 512-bits prime number. To achieve the same security level, we used the ECC group on Koblitz elliptic curve $y^2=x^3+ax^2+b$, which is defined on $F_{2^{163}}$ with $a=1$ and b is a 163-bit random prime. The running times are listed in **Table 1**.

There are two signature schemes with multi-signer and multi-verifier [18,19]. First, we compare our scheme with schemes [18,19] in **Table 2**. Then, we compare our scheme with other several signature schemes with multi-verifier in Table 3. To facilitate the comparison, we let $n=m=10$ in **Table 2** and let $n=1, m=10$ in **Table 3**.

We use a simple method to evaluate the computation cost. For example, Chang et al.'s [16] scheme needs $2n+1$ pairing-based scalar multiplication operations and $m+1$ pairing operations. So the resulting computation time is $3.76\times(2n+1)+11.68\times(m+1)=7.52n+11.68m+15.44$. Let $n=m=10$, then the resulting computation time is $(7.52+11.68)\cdot 10+15.44=207.44$

Follow on, we evaluate the length of signature. In our scheme, each signature contains 4 points of elliptic curve $y^2=x^3+ax^2+b$, thus the signature size is $\lceil 163\times 4\rceil/8=82$ byte.

In the schemes [11,17,18,19], each signature consists of 2 points of elliptic curve $E/F_p : y^2 = x^3 + x$, thus the signature size is $\lceil 512 \times 2 \rceil / 8 = 128$ byte. In the scheme [13], each signature contains 4 elements of finite fields $F_{2^{163}}$ and 2 points of elliptic curve $y^2 = x^3 + ax^2 + b$, thus the signature size is $\lceil 163 \times 6 \rceil / 8 = 123$ byte

Based on the above parameter and ways, the detailed comparison results are illustrated in **Table 2** and **Table 3**.

Table 1. Cryptographic operation time (in milliseconds).

P	E_G	E_S
11.68	3.76	0.61

Table 2. Comparison of our scheme with other schemes with multi-signer and multi-verifier

Scheme	Sign	Verify	Execution time ($n = m = 10$)	SL	Public key
Chang[18]	$2nE_G + P$	$E_G + mP$	207.44	128	ID-based
Ming[19]	$(3n+2)\ E_G + P$	$2mE_G + 2P$	230.56	128	Traditional public key
Our scheme	$(2n+m+2)E_S$	$(n+2m+2)\ E_S$	39.04	82	Certificateless

Table 3. Comparison of our scheme with several schemes with a signer and multi-verifier (Namely $n = 1$)

Scheme	Sign	Verify	Execution time ($m = 10$)	SL	Public key
Yang[11]	$4E_G + P$	$2mE_G + 2P$	125.28	128	Traditional public key
Zhang[13]	$(m+3)E_S$	$(m+4)E_S$	16.47	123	Traditional public key
Seo[17]	$3E_G + P$	$5mE_G + (m+2)P$	351.12	128	ID-based
Our scheme	$(m+4)E_S$	$(2m+3)\ E_S$	22.57	82	Certificateless

6. Conclusion

Although some good results have been achieved in speeding up the computation of pairing function in recent years, the computation cost of the pairing is much higher than that of the scalar multiplication over the elliptic curve group. Some designated multi-verifier signatures have been studied extensively over the last few years, however, there is not any work published on certificateless designated multi-verifier signatures. In this paper, we proposed the first certificateless multi-signer universal designated multi-verifier signature scheme and proved the security in the random oracle model. In our scheme, a set of signers cooperatively generate a public verifiable signature, the signature holder then can propose a new signature such that only the designated set of verifiers can verify it. Our scheme does not use pairing operation and is more efficient than previous ones. Due to the good properties of our scheme, it should be useful for practical applications.

7. Acknowledgment

The authors are grateful to the anonymous referees for their helpful comments and suggestions. This research is supported by the National Natural Science Foundation of China under Grants 61562012, the Innovation Group Major Research Projects of Department of Education of Guizhou Province under Grant No.KY[2016]026.

References

[1] A. Shamir, “Identity-based cryptosystems and signature schemes,” *Advances in Cryptology-Crypto 1984*, LNCS, vol. 196, pp. 47-53, 1984. Article (CrossRef Link)

[2] S.S. Al-Riyami, K.G.Paterson, “Certificateless public key cryptography,” *Advances in Cryptology-Asiacrypt 2003,* LNCS, vol. 2894, pp. 452-473, 2003. Article (CrossRef Link)

[3] M. Jakobsson, K. Sako, R. Impagliazzo, “Designated verifier proofs and their applications,” *Advances in Cryptology-Eurocrypt 1996*, LNCS, vol. 1070, pp. 142-154, 1996. Article (CrossRef Link)

[4] S. Saeednia, S. Kremer, O. Markowitch, “An efficient strong designated verifier signature scheme,” in *Proc. of International Conference on Information Security and Cryptology(ICISC 2003),* LNCS, vol. 2869, pp. 40-54, 2003. Article (CrossRef Link)

[5] R. Steinfeld, L. Bull, H. Wang, and J. Piperzyk, “Universal designated-verifier signatures,” *Advances in Cryptology-Asiacrypt 2003,* LNCS, vol. 2894, pp. 523-543, 2003. Article (CrossRef Link)

[6] F. Laguillaumie, D. Vergnaud, “Multi-designated verifiers signatures,” in *Proc. of International Conference on Information and Communications Security (ICICS 2004)*, LNCS, vol. 3269, pp. 495-507, 2004. Article (CrossRef Link)

[7] F. Laguillaumie, D. Vergnaud, “Multi-designated verifiers signatures: Anonymity without encryption,” *Information Processing Letters*, vol.102, no.2-3, pp. 127-132, 2007. Article (CrossRef Link)

[8] C. Willy, W. Susilo, and Y. Mu, “Universal designated multi verifiers signature schemes,” in *Proc. of 11th International Conference on Parallel and Distributed Systems (ICPADS 2005)*, pp. 305-309, 2005. Article (CrossRef Link)

[9] Y. Zhang, J. Zhang, and Y. Zhang, “Multi-signers strong designated verifier signature scheme,” *International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2008),* pp. 324-328, 2008. Article (CrossRef Link)

[10] G. Shailaja, K.P. Kumar, and A. Saxenh, “Universal designated multi verifier signature without random oracles,” in *Proc. of 9th International Conference on Information Technology (ICIT 2006),* pp. 168-171, 2006. Article (CrossRef Link)

[11] Y. Ming, Y. Yang, “Universal designated multi verifier signature scheme without random oracles,” *Wuhan University Journal of Natural Sciences*, vol.13, no.6, pp. 685-691, 2008. Article (CrossRef Link)

[12] T. Roeder, R.Pass, and F. Schneider, “Multi-verifier signatures,” *Journal of. Cryptology*, vol. 25, pp, 310–348, 2012. Article (CrossRef Link)

[13] Y. Zhang, M. Au, G. Yang, and W. Susilo, “(Strong) multi-designated verifiers signatures secure against rogue key attack,” in *Proc. of International Conference on Network and System Security on Network and System Security(ICNSS 2012),* LNCS, vol. 7645, pp. 334-347, 2012. Article (CrossRef Link)

[14] F. Zhang, W. Susilo, Y. Mu, and X. Chen, “Identity-based universal designated verifier signatures,” in *Proc. of International Conference on Embedded and Ubiquitous Computing (ICEUC 2005)* , LNCS, vol. 3823, pp. 825-834, 2005. Article (CrossRef Link)

[15] B. Kang, C. Boyd, and E. Dawson, “A novel identity-based strong designated verifier signature scheme,” *Journal of Systems and Software*, vol.82, no.2, pp. 270-273, 2009. Article (CrossRef Link)

[16] Lin, H.Y., “Toward secure strong designated verifier signature scheme from identity-based system,” *The International Arab Journal of Information Technology*, vol. 11, no. 4, pp.315-321, 2014. Article (CrossRef Link)

[17] S.H. Seo, J.Y. Hwang, K.Y. Choi, and D.H. Lee, “Identity-based universal designated multi-verifiers signature schemes,” *Computer Standards and Interfaces*, vol. 30, no.5, pp.288-295, 2008. Article (CrossRef Link)

[18] T. Y. Chang, “An ID-based multi-signer universal designated multi-verifier signature scheme,” *Information and Computation*, vol. 209, pp. 1007-1015, 2011. Article (CrossRef Link)

[19] Y. Ming, Q. Jin, and X. Zhao, “A multi-signer universal designated multi-verifier signature scheme in the standard model,” *Journal of Computational Information Systems*, vol.9, no. 9, pp.3751-3758, 2013. Article (CrossRef Link)

[20] X. Huang, W. Susilo, Y. Mu, and F. Zhang, “Certificateless designated verifier signature schemes,” in *Proc. of the 20th International Conference on Advanced Information Networking and Applications*, pp. 15-19. 2006. Article (CrossRef Link)

[21] Y. Ming, X. Shen, and Y. Wang, “Certificateless universal designated verifier signature schemes,” *The Journal of China Universities of Posts and Telecommunications*, vol.14, no.3, pp, 85-90, 2007. Article (CrossRef Link)

[22] B.Yang, Z. Hu, and Z. Xiao, “Efficient certificateless strong designated verifier signature scheme,” in *Proc. of 2009 International Conference on Computational Intelligence and Security*, pp. 432-436, 2009. Article (CrossRef Link)

[23] H.Du, Q. Wen, “Efficient certificateless designated verifyer signatures and proxy signatures,” *Chinese Journal of Electronics*, vol.18, no.1, pp, 95-100, 2009. Article (CrossRef Link)

[24] D. He, J. Chen, “An efficient certificateless designated verifier signature scheme,” *International Arab Journal of Information Technology*, vol.10, no.4, pp. 389-396, 2013. Article (CrossRef Link)

[25] M. Bellare, G. Neven, “MultiSignatures in the plain public key model and a general forking lemma,” in *Proc. of the 13th ACM Conference on Computer and Communications Security*, pp. 390-399, 2006. Article (CrossRef Link)

Lunzhi Deng received his B.S. from Guizhou Normal University, Guiyang, PR China, in 2002; M.S. from Guizhou Normal University, Guiyang, PR China, in 2008; and Ph.D. from Xiamen University, Xiamen, PR China, in 2012. He is now a professor in the School of Mathematics and Computer Science, Guizhou Normal University, Guiyang, PR China. His recent research interests include algebra and information safety.

Yixian Yang is a professor of Beijing University of Posts and Telecommunications, Beijing, PR China. He is a member of the China Science and Technology Commission of the Ministry of Education, has been published more than 300 papers in the IEEE Trans. On AES, IEEE Trans. On Comm., IEEE Trans. On EMC and Discrete Applied Mathematics and other international most authoritative academic journals.

Yuling Chen received her BS from Taishan University, Taian, PR China, in 2006; MS from Guizhou University, Guiyang, PR China, in 2009. She is now a associate professor in Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, PR China. Her recent research interests include cryptography and information safety.