An Efficient Heterogeneous Signcryption Scheme from Certificateless to Identity-based Cryptosystem

Shufen Niu[1], *Zhenbin Li*[1,*], Miao Tian[1] , Caifen Wang[1] and Xiangdong Jia[1]

[1] College of Northwest Normal University of Computer Science and Engineering , Lanzhou, 730070, PR China

Abstract. The heterogeneous signcryption can not only realize the confidentiality and un-forgeability of the information transmission between different public key cryptography environments, but also reduce the communication cost. The proposed scheme uses bilin-ear pairings between certificateless cryptography and identity-based cryptography to construct a heterogeneous signcryption scheme. In addition, two cryptography systems use different secret master keys. Furthermore, the proposed scheme has the indistin-guishability against adaptive chosen ciphertext attacks and existential unforgeability against adaptive chosen-message attacks in the random oracle model. Through the data analysis, this scheme is more effective than other similar types of scheme.

Keywords. Heterogeneous signcryption, Certificateless cryptography, Identity-based cryptography, Different secret master, Secure

1 Introduction

Data confidentiality and authentication should be ensured when transmitting data in public networks. The traditional approach, signature then encryption involves numerous calculations, costly communication, and low computational efficiency. To improve the computational efficiency, Zheng[1] proposed the notion of signcryption in 1997. Signcryption can simultaneously perform signature and encryption, which enhances the efficiency of communication systems.

In the future, heterogeneous network technology is crucial to secure communication. Generally, different cryptosystems should be used in heterogeneous networks. Heterogeneous Signcryption (HSC) should be explored to ensure the confidentiality and authentication of messages in a heterogeneous network.

In 2010, Sun[2]. proposed the HSC scheme that communicates from Traditional Public Key Instruction (TPKC) to Identity-based Public Key Cryptography (IDPKC)[3]. However, the scheme only addresses external security threats[4]. In 2011, Huang et al.[5] proposed a HSC scheme with key privacy, which satisfies internal security[4]. In 2013, Fu et al.[6] proposed an IDPKC-to-TPKC construction of a multi-receiver signcryption scheme. In the same year, Li et al. [7] proposed a two-way HSC scheme. However, these HSC schemes only consider the signcryption problem between IDPKC and TPKC. Although these schemes use different public key cryptosystems, the sender and the receiver use the same system parameters. In 2016, Zhang et al.[8] proposed a HSC scheme that communicates from Certificateless Public Key Cryptography (CLPKC) [9, 10] to TPKC. To the best of our knowledge, only four HSC schemes have been developed, and the signcryption problem between IDPKC and CLPKC has not been addressed in the literature.

Existing heterogeneous signcryption schemes use similar master secret keys. In 2016, Li et al.[11] proposed a multi-receiver signcryption scheme for heterogeneous systems. Their work provided inspiration for the current study.

Based on Li[12], the proposed scheme develops a scheme from CLPKC[13] to IDPKC. The present paper introduces a formal security model called Efficient CLPKC-to-IDPKC Heterogeneous Signcryption Scheme and proves the model's semantic security and existential unforgeability in the random oracle model [14]. Our scheme is more secure and efficient than other existing schemes. The proposed scheme has the following features.

- This scheme uses different master secret keys in different cryptosystem systems, and improve the security of the system.
- Compared with existing heterogeneous schemes, this scheme is reduced system uptime through reducing the number of bilinear pairs. This scheme is more secure and efficient.
- In the random oracle model, this scheme ensures the confidentiality and un forge ability of data in the context of the Variants Decisional Bilinear Dif-fie-Hellman Problem (VDBDHP) ,Variants Computational Bilinear Dif-fie-Hellman Problem (VCBDHP) and Discrete Logarithm Problem (DLP).

The rest of this paper is organized as follows. Related knowledge are reviewed in Section 2.The formal

* Corresponding author: 775627945@qq.com

definition and security model of the proposed scheme are described in Section 3. The heterogeneous signcryption scheme is presented in Section 4. The security analysis of the scheme is discussed in Section 5. The performance analysis is discussed in Section 6. Finally, Section 7 concludes the paper.

2 Related knowledge

This section briefly describe bilinear maps and hard problems.

Let G_1 be a cyclic additive group generated by P, with a prime order q, and G_2 be a multiplicative group of the same order. The bilinear pairing is a map $e: G_1 \times G_1 \to G_2$ with the following properties.

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

• Non-degeneracy: $P,Q \in G_1$ such that $e(P,Q) \neq 1$.

• Computability: An efficient algorithm is available for computing $e(P,Q)$ for any $P,Q \in G_1$.

The hard problems involved in the security of the proposed scheme are described as follows.

2.1. Hard Problems

• Variants Decisional Bilinear Diffie-Hellman (VDBDH) problem

Given $(P,aP,bP,cP,dP,c^{-1}P,d^{-1}P)$ unknown $a,b,c,d \in Z_q^*$, compute for $e(P,P)^{abc^{-1}}$.

•Variants Computational Bilinear Diffie-Hellman (VCBDH) problem

Given $(P,aP,bP,cP,dP,c^{-1}P,d^{-1}P)$ with unknown $a,b,c,d \in Z_q^*$ compute for $e(P,P)^{abd^{-1}}$.

• Discrete Logarithm Problem (DLP)

Given a cyclic additive group G and generator P, the DLP is defined that given (P,aP), compute $a \in Z_q^*$.

According to the existing schemes [15,16], it is easy to known that both VDBDHP and VCBDHP are hard. Obviously, Discrete Logarithm Problem (DLP) is hard.

3 Semantic security and security model

3.1. Semantic security

A CLC $\to$ IBC heterogeneous signcryption scheme generally includes the following algorithm:

• **Setup.** Choose a security parameter l to input. The PKG and KGC generate their own master key, and outputs the system's parameters p*arams*.

• **CLPKC-KG.** The algorithm runs by the KGC of the CLPKC system. Submit the identity ID_U and a secret value x_U, the corresponding sk_s is generated as the private key of the CLPKC user.

• **IDPKC-KG.** The algorithm runs by the PKG of the IDPKC system. By inputting a user's identity ID_U, the corresponding private key S_U is generated.

• **Signcryption.** Given the sender's identity ID_s, the receiver's identity ID_r, the private key S_U and the message m, the sender runs the signcryption algorithm to generate a ciphertext σ.

• **Unsigncryption.** Given the sender's identity ID_s, the private key S_U of the receiver and a ciphertext σ, the receiver computes and outputs the plaintext m, or the symbol $\perp$.

3.2. Security Models

3.2.1 Confidentiality

A heterogeneous signcryption scheme has the IND-CCA2 property if no probabilistic polynomial time adversary A has a non-negligible advantage in the following game:

Setup. Challenger C runs the setup algorithm with a security parameter l and send system parameters and public keys to the A, while the master key is kept secret.

Phase 1. A can ask several kinds of queries to the following random oracles.

• **Key generation query.** A submits an identity ID_U, C runs the IDPKC-KG algorithm to generate the private key S_U and send it to A.

• **Unsigncryption query.** A submits a signcrypted message σ and an identity ID_s. C computes the private key S_U. If σ is a valid ciphertext, C returns a message $m = Unsigncrypt(\sigma, params, S_U)$; otherwise, an error symbol $\perp$ is returned.

Challenge. A decides when Phase 1 ends. A selects two plaintexts m_0 and m_1 of the same length, a sender's identity ID_A, and a receiver's identity ID_B^*, which it wants to challenge. A is not allowed to ask for the private key to ID_B in the first stage. C then selects $b \in \{0,1\}$ and runs the corresponding algorithms to obtain the ciphertext σ. σ is transmitted to A.

Phase 2. A can perform the queries similar to those in Phase 1. A cannot query the key extraction for the target identities. A should not query the Unsigncryption for σ.

Guess. Finally, A produces a bit b', A wins the game if $b' = b$.

The advantage of A is defined as $Adv(A)=|\Pr[b'=b]-1|$,where Pr[b'=b] denotes the probability that $b' = b$.

3.2.2 Unforgeability I

A CLPKC→IDPKC heterogeneous signcryption scheme is referred to as EUF-CMA-I. The game is defined between a challenger C and an adversary F_I. If no

probabilistic polynomial time adversary F_I has a non-negligible advantage against a challenger in the following game:

Setup. The challenger C runs the setup algorithm. It sends public parameters $params$ to the adversary F_I, while the master keys are kept secret. When F_I receives $params$, F_I outputs a target identity ID^*.

Attack. F_I issues several kind of queries.

• **Extract partial private key query.** F_I selects an identity ID_i, C computes the partial private key S_i, and transmits it to F_I.

• **Extract secret key query.** F_I produces an identity ID_i, C computes the se cret key x_i, and transmits it to F_I.

• **Request public key query**. When F_I receives a public key extraction query for an identity ID_i, C runs the CLPKC-KG algorithm to the public key pk_i, and transmits it to F_I.

• **Replace public key query.** For any identity ID_i and a valid public key pk_i', C replaces pk_i with a value pk_i'.

• **Signcrypt query.** F_I produces a message m, a sender's identity ID_s, and a re ceiver's identity ID_r, C first finds the sender's private key sk_s from the CLPKC-KG algorithm. C then runs signcryption algorithm and sends σ to F_I.

Forgery. F_I produces a tuple $(ID_s^*, \sigma^*, ID_r^*)$. ID_s^* cannot be queried a partial private key extraction. F_I wins if the unsigncryption returns m.

The advantage of F_I is defined as the probability that F_I wins the EUF-CMA-I game.

3.2.3 Unforgeability II

A CLPKC→IDPKC heterogeneous signcryption scheme is referred to as EUF-CMA-II , The game is defined between a challenger C and an adversary F_{II}. If no probabilistic polynomial time adversary F_{II} has a non-negligible advantage against a challenger in the following game:

Setup. The challenger C runs the setup algorithm. It generates master secret keys s_{KGC}, s_{PKG} and system's public parameters $params$ to the adversary F_{II} while s_{PKG} is kept secret. When C receives the public parameters, F_{II} outputs a challenge identity ID^*.

• **Extract secret key query.** F_{II} produces an identity ID_i, C computes the secret key x_i from the CLPKC-KG algorithm, and transmits it to F_{II}.

• **Request public key query.** When F_{II} receives a public key extraction query for an identity ID_i, C runs the CLPKC-KG algorithm to the public key pk_i, and transmits it to F_{II}.

• **Signcrypt query.** F_{II} produces a message m, a sender's identity ID_s and a re ceiver's identity ID_r. C first finds the sender's private key sk_s from the CLPKC-KG algorithm. C then runs the signcryption algorithm and sends σ to F_{II}.

Forgery. F_{II} produces a tuple $(ID_s^*, \sigma^*, ID_r^*)$. ID_s^* cannot be queried a partial private key extraction. F_{II} wins if the unsigncryption returns m. σ was not produced by any signcrypt query.

The advantage of F_{II} is defined as the probability that F_{II} wins the EUF-CMA-II game.

4 Describe the scheme

The heterogeneous signcryption scheme is described in this section. It can be allowed as the sender in the CLPKC system and as the receiver in the IDPKC system. The proposed scheme is described as follows.

Setup. In the heterogeneous system, let G_1 be a cyclic additive group of prime order q and G_2 be a cyclic multiplicative group of the same order. Let P be the generator of G_1. A bilinear map is defined as $e: G_1 \times G_1 \to G_2$. Three cryptographic hash functions are chosen,namely, $H_1: \{0,1\}^* \to G_1$, $H_2: G_2 \to \{0,1\}^{l_m}$ and $H_3: \{0,1\}^* \to Z_q^*$. where l_m is the length of a message. The KGC selects $s_1 \in Z_q^*$ as the master secret key and sets $P_1 = s_1 P$ as the public key of the CLPKC system . The PKG selects $s_2 \in Z_q^*$ as the master secret key and sets $P_2 = s_2 P$ as the public key of the IDPKC system . The global public parameters are $param = \langle G_1, G_2, e, P, P_1, P_2, H_1, H_2, H_3 \rangle$ and the master secret key are s_1 and s_2.

CLPKC-KG. As a sender in the CLPKC system, it uses an identity ID_A and com putes the private key using the following algorithm.

• Compute $Q_A = H_1(ID_A)$ and set the partial private key $S_A = s_1^{-1} Q_A$.

• Choose a random secret value $x_A \in Z_q^*$.

• Compute the public key $pk_A = x_A P$ and the private key $D_A = x_A S_A$.

IDPKC-KG. The PKG computes the private key of an identity ID_B as $S_B = s_2^{-1} Q_B$ where $Q_B = H_1(ID_B)$.

Signcryption. The system parameters, namely, the message m of length l_m, the receiver's identities ID_B are given, Then the sender uses the following algorithm:

• Choose a random number $r \in Z_q^*$ and compute $U_1 = rP_2$, $U_2 = rQ_A$.

• Compute $R=e(P,Q_B)^r$ and $k=H_2(R)$.

• Compute $c=m\oplus R, h=H_3(m,k,U_1,U_2), V=(r+h)D_A$.

• Output ciphertext as $\sigma=(c,U_1,U_2,V)$.

Unsigncryption. When the system receives a ciphertext $\sigma=(c,U_1,U_2,V)$ and the system parameters, the receiver with an identity ID_B unsigncrypts the ciphertext σ as follows:

• Compute $R=e(U_1,S_B)$ and $k=H_2(R)$.

• Recover $m=c\oplus R$.

• Compute $h=H_3(m,k,U_1,U_2)$.

•Accept the message if and only if the equation holds: $e(P_1,V)=e(pk_A,U_2+hQ_A)$.

Correctness. The following equations show the correctness of the scheme.

When receiving the ciphertext, the receiver can decrypt the ciphertext as follows:

$$R=e(U_1,S_B)=e(rP_2,s_2^{-1}Q_B)=e(P,Q_B)^r$$

Simultaneously, the receiver can verify the following equalities to show:

$$e(P_1,V)=e(s_1P,(r+h)D_A)=e(s_1P,(r+h)x_AS_A)=e(x_AP,(r+h)s_1S_A)=e(pk_A,(r+h)Q_A)=e(pk_A,rQ_A+hQ_A)=e(pk_A,U_2+hQ_A)$$

The scheme can be modified into environments with different systems parameters. For example, the PKG has system parameters $\langle G_1^*,G_2^*,P^*,H_1,H_2,H_3\rangle$ and the KGC has the system parameters $\langle G_1,G_2,P,H_1,H_2,H_3\rangle$.The scheme will remain secure.

5 Security Analysis

This section will show the scheme's security analysis, including confidentiality and unforgeability.

5.1. Confidentiality

Theorem 1. In the random oracle, if an IND-CCA2 adversary C can attack our proposed scheme, then there exists an algorithm C that can solves the VDBDH problem.

Proof. The distinguisher C inputs $(P,aP,bP,cP,dP,c^{-1}P,d^{-1}P,\theta)$ and attempts to decide whether $\theta=e(P,P)^{abc^{-1}}$ or not. C will run A as a subroutine and act as $A's$ challenger in the IND-CCA2 game. We model H_i ($i=1,2,3$) as random oracles. C keeps three lists L_1,L_2,L_3 to store the answers.

Setup. At the beginning of the game, C runs the setup algorithm and set $P_b=cP$ and provides the system parameters to attacker A.

Phase 1. A can request several queries. To respond to these queries, H_1,H_2,H_3 are random oracles controlled by C as follows.

• H_1 **-query**: C selects a random number $j\in\{1,2,\cdots,qH_1\}$. A asks for a polynomial bounded number of H_1 queries on identities of his choice. At the $j-th$ H_1 query, C answers by $H_1(ID_j)=bP$. For queries $H_1(ID_i)$ with $i\neq j$, C selects $e_i\in Z_q^*$, puts the pair (ID_i,e_i) in list L_1 and answers $H_1(ID_i)=e_iP$.

• H_2,H_3 **-queries**. When A asks queries on these hash values, C checks the list. If an entry for the query is found, it will be return to A. Otherwise, a random value is chosen to send to A, and the answer will then be stored in the list.

• **Key extraction query.** When A asks a key extraction query in identity ID_i, if $ID_i=ID_j$, then C fails and stops. If $ID_i\neq ID_j$, C finds (ID_i,e_i) from H_1 (which indicates that C previously answered $H_1(ID_i)=e_iP$ on an H_1 query on ID_i). C updates the list, and then computes the corresponding private key $e_ic^{-1}P$. C returns it to A.

•**Unsigncryption queries.** For an unsigncryption query in a ciphertext σ' for iden tities ID_A and ID_B. C follows the steps below:

Case 1: $ID_B=ID_j$. C always answers A that σ' is invalid. If σ' is valid from $A's$ viewpoint, C will fail.

Case 2: $ID_B\neq ID_j$. C computes $R'=e(U_1',S_B)$ (C could obtain S_B from the key extraction algorithm because $ID_B\neq ID_j$). C then runs the H_2 simulation algorithm to obtain $k'=H_2(R')$ and computes $m'=c\oplus R'$.

Finally, C runs the H_3 simulation algorithm to generate $h'=H_3(m',k',U_1',U_2')$ and checks if $e(P_1,V')=e(pk_A,U_2'+h'Q_A)$ holds. If the above equation does not hold, C returns $\perp$. Otherwise, C returns m'.

Challenge. A outputs two plaintexts m_0,m_1 to C. C randomly chooses $b\in\{0,1\}$ and computes the signcryption of message m_b as follows.

• It sets $U_1^*=aP$, obtains $k^*=H_2(\theta)$ (where θ is C candidate for the VDBDHP).

• Compute $c_b=m_b\oplus R$.

• Randomly choose $x^*,h^*\in Z_q^*$.

• Compute $U_2^*=-h^*Q_A+x^*pk_A$ and $V^*=x^*P$.

• Adds (c^*,U_1^*,U_2^*,V^*) to list L_3.

Finally, C produces the challenged ciphertext $\sigma^*=(c_b,U_1^*,U_2^*,V^*)$ and send it to A.

Phase 2. A performs new queries that are treated the same way as in Phase 1.

Guess. A produces a bit b' as its guess. At the moment, if $b=b'$, C outputs $\theta=e(U_1^*,S_{ID_j})=e(aP,c^{-1}bP)=e(P,P)^{abc^{-1}}$ as a solution of the VDBDHP. Otherwise, C stops and outputs "failure".

5.2 Unforgeability I

Theorem 2. In the random oracle model, if an EUF-CMA-I adversary F_I can attack our proposed scheme, then there exists an algorithm C that can solve the VCBDH problem.

Proof. If an EUF-CMA-I adversary F_I exists, then we construct a simulator C that uses F_I to compute $e(P,P)^{abd^{-1}}$ from an instance $(P,aP,bP,cP,dP,c^{-1}P,d^{-1}P)$ of the VCBDHP.

Setup. The challenger C runs the setup algorithm. It sets $P_1 = dP$, and sends the system parameters $params = \{G_1, G_2, e, P_1, P_2, H_1, H_2, H_3\}$ to F_I , then F_I outputs a challenge identity ID^* .

• H_1 -query. When C receives a query $(ID_j, \alpha_j, H_1(ID_j), D_j)$. If $(ID_j, \alpha_j, H_1(ID_j), D_j)$ exists in L_1 , then h_j is returned to F_I . Otherwise, C does as follows:

(1) If $ID_j \neq ID^*$, select $\alpha_j \in Z_q^*$ at random, then compute $H_1(ID_j) = \alpha_j P$, $D_j = \alpha_j P_1$;otherwise, set $\alpha_j = D_j = \perp$, $H_1(ID_j) = bP$.

(2) Put $(ID_j, \alpha_j, H_1(ID_j), D_j)$ into L_1 , return $H_1(ID_j)$.

• $H_i (i = 2,3)$ **-query.** When C receives a query, if the corresponding query exists in $L_i (i = 2,3)$, C returns it to F_I . Otherwise, C randomly chooses an number as the query answer and return to F_I . Meanwhile, C puts the query result into L_i .

• **Public key query.** When C receives a query (ID_j, x_j, pk_j) . If (ID_j, x_j, pk_j) exists in the public key-list, then pk_j is returned to F_I , Otherwise, C chooses $x_j \in Z_q^*$ at random, computes $pk_j = x_j P$, then puts (ID_j, x_j, pk_j) into the public key-list and return pk_j as answer.

• **Extract secret key query.** When C receives a query ID_j . If C replaces the public key of ID_j , then it returns $\perp$. Otherwise, (ID_j, x_j, pk_j) exists in the key-list, and C returns x_j as answer.

• Extract partial private key query. When C receives a query (ID_j, x_j, D_j) , C does as follows:

(1) If $ID_j = ID^*$,then C aborts; if $(ID_j, \alpha_j, H_1(ID_j), D_j)$ exists in L_1 , D_j is returned to F_I ; otherwise, C first makes a query to H_1 .

(2) Put (ID_j, x_j, D_j) into L_1 ; return D_j .

• **Replace public key query.** When C receives a replace public key query (ID_j, pk_j') , C first finds (ID_j, x_j, pk_j) on the public key-list, then C updates the public key-list with tuple $(ID_j, \perp, pk_j')$, and sets $x_j = \perp$, $pk_j = pk_j'$.

• **Signcryption query.** When C receives a signcryption query, let ID_A , ID_B be the identies of the sender and receiver, respectively, m be the plaintext, C will do as follows:

(1) If $ID_A \neq ID_j$. C computes the private key S_{ID_A} corresponding to ID_A by running the key extraction query algorithm. Then C answers the query.

(2) If $ID_A = ID_j$, C simulates the signcryption algorithm to create a signcryption in the following steps. C chooses $x, h \in Z_q^*$ randomly, and computes $U_1 = xP_b$, $R = e(U_1, S_B)$ (C could obtain S_B from the key extraction algorithm because $ID_B \neq ID_j$). C runs the H_2 simulation algorithm to find $k = H_2(R)$ and computes $c = m \oplus R$. Then C computes $U_2 = -hQ_A + xpk_A$ and $V = xP$, and adds (c, U_1, U_2, h) to list L_3 . If a collision happens, C outputs fail and exits. Otherwise, C gives the ciphertext σ to F_I .

Attack. F_I adaptively performs a polynomially bounded number of queries to the various oracles in this phase.

Forge. Eventually, F_I outputs a tuple $(m^*, \sigma^*, ID_s^*, pk_s^*)$. If $ID_s^* \neq ID^*$, C aborts. Otherwise, by forking lemma, C chooses different h, h^* and interacts with F_I the same random tape, then the adversary can give a different forger σ^* . σ^* and σ should verified by the equation $e(P_1, V) = e(pk_A, U_2 + hQ_A)$ and $e(P_1, V^*) = e(pk_A, U_2 + h^*Q_A)$. Where if $P_1 = dP$ and $pk_A = \beta P$, C can compute $bd^{-1}P = \dfrac{V - V^*}{\beta(h - h^*)}$, Simultaneously, C delivers the value of $e(P,P)^{abd^{-1}}$ as $e(aP, bd^{-1}P)$. Hence, C successfully solves the VCBDHP.

5.3 Unforgeability II

Theorem 3. In the random oracle model, if an EUF-CMA-II adversary F_{II} can attack our proposed, then there exists an algorithm C that can solve the Discrete Logarithm Problem (DLP).

Proof. If an EUF-CMA-II adversary F_{II} exists, then we construct a simulator C that uses F_{II} to compute $a \in Z_q^*$ from an instance $\langle P, aP \rangle$ of the DLP.

Setup. C first runs the setup algorithm to generate s_1, s_2 as the master keys, and sets $P_2 = s_2 P$, then gives s_1 , $params$ to the attacker F_{II} , s_2 is kept secret. F_{II} outputs a challenge identity ID^* .

• $H_i (1,2,3)$ **-query.** When C receives a query, if the corresponding query exists in H_i -list, C returns it to F_{II} . Otherwise, C randomly chooses an integer as the query result and returns it to F_{II} . Meanwhile, C puts the query result into the H_i -list.

• **Public Key query.** When C receives a query (ID_j, x_j, pk_j). If (ID_j, x_j, pk_j) exists in the public key-list, then C returns pk_j. Otherwise, C chooses $x_j \in Z_q^*$ at random, computes $pk_j = x_j P$, then puts (ID_j, x_j, pk_j) into the public key-list and returns pk_j as answer.

• **Signcryption query.** The same as Unforgeability I's Signcryption query.

Attack. F_{II} adaptively performs a polynomially bounded number of queries to the various oracles in this phase.

Forge. Eventually, F_{II} outputs a tuple $(m^*, \sigma^*, ID_s^*, pk_s^*)$. If $ID_s^* \neq ID^*$, C aborts; otherwise, C randomly chooses $h \in Z_q^*$, then the adversary can give a forger σ. We know that σ should verified by the equation $e(P_1, V^*) = e(pk_A^*, U_2^* + hQ_A^*)$. And C set $pk_A^* = aP$ and compute the equation, then C obtain the result $a = \frac{s_1 V^*}{U_2^* + hQ_A^*}$ as the solution to given DLP. Hence, C successfully solves the DLP.

6 Performance analysis

We compare the proposed scheme with other existing signcryption schemes [2],[7],[18]. The Table 1 shows the efficiency and distinction of the schemes. We denote Pa as the total number of pairing computations required, mark the Mu as the total number of point multiplications required and denote the Ex as the total number of exponentiations required. We compare our scheme with scheme [2],[7],[18] even though these schemes are all in the same cryptosystems. The results are shown in Table 1 and Table 2. As shown in Table 1 and Table 2, our scheme has a lower computation cost than other schemes.

Table 1. Comparison with existing scheme for signcryption.

Scheme	Pa	Mu	Ex	Master key
Sun[2]	1	3	1	Same
Li[7]	2	3	2	Same
F Li[18]	1	3	1	Same
Our scheme	1	2	1	Different

Table 2. Comparison with existing scheme for unsigncryption.

Scheme	Pa	Mu	Ex	Master key
Sun[2]	4	1	0	Same
Li[7]	3	2	2	Same
F Li[18]	5	0	0	Same
Our scheme	3	1	0	Different

In the simulation, we show the time by measuring the performance of the Signcryption and Unsigncryption. Our program is written in with the PBC library [17], and we run on a personal computer with 3.10GHz CPU and 2GB of RAM, using Linux operating system. In order to the effect of comparison, we will adopt Y Li et al [11]. Both schemes are set up in heterogeneous system, and both the schemes use different system parameters. We can get the result as Fig 1, which SC means signcryption operation and USC means unsigncryption operation. In the performance of the Signcryption and Unsigncryption, our scheme has been improved in terms of efficiency.

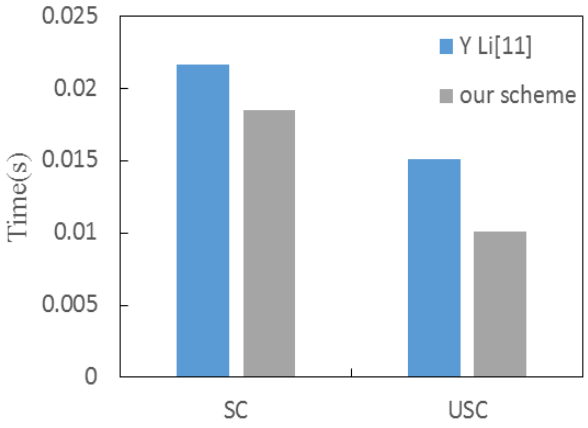

Fig. 1. Performance comparison between Li's and our schemes.

7 Conclusion

We have developed an efficient signcryption scheme for the CLPKC→IDPKC heterogeneous system. Compared with the existing signcryption schemes, the proposed scheme chooses different master secret keys in different systems. The security models are provided, and the proposed scheme ensures the confidentiality and unforgeability of data in the random oracle model. The proposed is more secure because it uses different master keys. Furthermore, the proposed scheme can be used in environments with different system parameters.

Acknowledgements

The work was supported by the National Natural Science Foundation of China under grant 61562077, 61462077, 61662071, 61662069, and Natural Science Foundation of Gansu Province for Distinguished Young Scholars 1308RJDA007.

References

1. Zheng Y. Digital signcryption or how to achieve cost in AdvancesinCryptology'97 on Springer, 1997. pp.165-179.
2. Sun, Li H.Efficient signcryption between TPKC and IDPKC and its multi-receiver construction, Science China Information Sciences, Vol 53, 2010, pp.557-566.
3. Shamir. Identity-Based cryptosystems and signature schemes, Springer, Heidelberg, 1985, pp.47-53.
4. An J H, Dodis Y, and Rabin T. On the Security of Joint Signature and Encryption, Springer, Berlin, 2002, pp.83-107.
5. Huang Q, Wong D S, and Yang G. Heterogeneous signcryption with key Privacy, Computer Journal, Vol 54, 2011, pp.525-536.
6. Fu, Li, and Liu W. IDPKC-to-TPKC construction of multi-receiver signcryption, Proceedings of the INCoS (5) on IEEE, 2013, pp.335- 339.
7. Li, Zhang H, and Takagi T. Efficient signcryption for heterogeneous systems, IEEE Systems Journal, Vol 7, 2013, pp.420-429.
8. Zhang et al, CLPKC-to-TPKI Heterogeneous Signcryption Scheme with Anonymity, Acta Electronica Sinica, Vol 44, 2016, pp.2432-2439.
9. Al-Riyami S S, and Paterson K G. Certificateless public key cryptography, Advances in Cryptology-Asiacrypt on Springer, 2003, pp.452-473.
10. Zhou Y W, Yang B, and Zhang W Z. Provably secure and efficient certificateless generalized signcryption, Chinese Journal of Computers, Vol 39, 2016, pp.543-551.
11. Li Y, Wang C, Zhang Y, and Niu S. Privacy preserving multi‐receiver signcryption scheme for heterogeneous systems, Security & Communication Networks, Vol 9, 2017, pp.4574-4584.
12. Li F, Shirase M, and Takagi T. Efficient Multi-PKG ID-Based Signcryption for Ad Hoc Networks, Information Security and Cryptology on Springer, 2009, pp.289-304.
13. Al-Riyami, S. S, Paterson, and K. G. Certificateless public key cryptography. International Conference on the Theory and Application of Cryptology, Springer, Berlin, 2003, pp.452-473.
14. Bellare, M. and Rogaway P. Random Oracles are Practical, 1993, pp.366–368.
15. Bao F, Deng R H, and Zhu H F. Variations of Diffie-Hellman Problem, in Information and Communications Security, Huhehaote, China, 2003, pp.301-312.
16. Chow S S M, Yiu S M, and Hui L C K. Efficient Forward and Provably Secure ID-Based Signcryption Scheme with Public Verifiability and Public Ciphertext Authenticity, Vol 2971, 2004, pp.352-369.
17. PBC Library. http://crypto.stanford.edu/pbc/.
18. Li, Fagen, Y. Han, and C. Jin. Practical access control for sensor networks in the context of the Internet of Things , Computer Communications, Vol 89,2016, pp.154-164.