Wireless Pers Commun (2015) 85:1727–1739
DOI 10.1007/s11277-015-2864-6

On security of a Certificateless Hybrid Signcryption Scheme

Aihan Yin[1] · **Hongchao Liang**[1]

Published online: 30 June 2015

Abstract Certificateless hybrid signcryption is a newly cryptosystem that plays a great role in some storage-constrained networks when confidentiality and authenticity are needed simultaneously. Now considering almost all certificateless signcryption schemes that have been proposed in the literature cannot effectively against the public-key-replacement attacks. In this paper, we proposed a hybrid signcryption scheme in the certificateless setting to fill this security gaps, and its security has been verified to achieve the confidentiality and unforgeability in random oracle model. Moreover, performance analysis shows the proposed scheme is efficient and practical.

Keywords Certificateless hybrid signcryption · Bilinear pairing · Provable security · Public-key-replacement attacks

1 Introduction

Confidentiality, integrity, non-repudiation and authentication are the important requirements for many cryptographic applications. In the traditional Public Key Cryptography (PKC), when a sender wishes to transmit a message to the receiver, firstly the sender needs a trusted Certificate Authorities (CA) to issue digital certificate and authenticated public key [1]. And certificate is notoriously considered to be costly to use and manage. Observing the heavy overheads of storing, transferring and verifying the certificate in the traditional PKC [2]. To simplify certificate management procedures of traditional PKI, identity-based cryptography (IBC) was proposed by Shamir [3] and makes deployment practical. The idea of IBC is to get rid of certificates by allowing the user's public key to

✉ Hongchao Liang
mr_lianghc@163.com

[1] Department of Information Engineering, East China Jiaotong University, Shuanggang Road 808, Nanchang 330013, Jiangxi, China

some unique message that identifies a user in the system. Examples of such message include IP addresses and email addresses. This eliminates the requirement to link the public key with a user, and several practical IBC schemes have been devised [4–6]. However, the private key corresponding to a user's public key is derived by a trusted authority called the private key generator (PKG), which leads to the private key escrow problem.

To solve this problem, Al-Riyami and Paterson [7] proposed a new cryptographic paradigm called Certificateless Public Key Cryptography (CL-PKC). In this cryptosystem, a user private key is a combination of some user-chosen secret and some contribution of a trusted PKG, in such a way that the key escrow problem can be solved. In order to perform encryption and signature simultaneously more efficiently in some network environments e.g. smartcards or wireless sensor networks (WSN) etc. Barbosa and Farshim [8] proposed a novel notion of certificateless signcryption (CLSC) in 2008, which is combined with the signcryption technology that simultaneously fulfills both the functions of public key encryption and digital signature in a logically single step. Subsequently, as one of the research hotspots, the CLSC has attracted extensive attention from the academia.

Recently, many efficient CLSC schemes have been proposed [9–12]. Wu et al. [10] proposed an efficient CLSC scheme in 2008, which requires four pairing operations in the signcryption and unsigncryption phase, but unfortunately it was found insecure by Selvi et al. [11]. And in 2011, Selvi et al. [12] also introduced a new security CLSC scheme, which requires five pairing operations in same phase. While above schemes are required to send message from a particular collection.

A practical way to perform secrecy communication for large messages is to use hybrid encryption that separates the encryption into two parts: one part uses public key techniques to encrypt a one-time symmetric key; the other part uses the symmetric key to encrypt the actual message. In such a construction, the public key part of the algorithm is known as the key encryption mechanism (KEM) while the symmetric key part is known as the data encapsulation mechanism (DEM). The formal treatment of this paradigm originates in the work of Shoup and Cramer [13]. And the resulting hybrid encryption paradigm has received much attention in recent years [14–16]. It is very attractive as it gives a clear separation between the various parts of the cipher allowing for modular design.

In 2013, Li et al. [17] extended the concept of hybrid signcryption to CLSC cryptosystem, which can handle message of arbitrary length, but it still requires six pairing operations. In 2014, Zhou et al. [18] introduce a certificateless generalized signcryption scheme in formal security model to against the malicious-but passive key generation center attacks, but its compute overhead is still high in the signcryption and unsigncryption phase.

In this paper, aimed at designing an efficient protocol for storage-constrained network. We proposed a practical and provably secure certifiacteless hybrid signcryption scheme, which used the technique of the public key binding in the extract-partial-private-key and only required four bilinear pairing operations. Therefore, this scheme not only can resist efficiently the current known security attacks, especially against public-key-replacement attacks, but also has lower communication overhead and shorter ciphertext length.

The paper is organized as follows. In Sect. 2, the preliminaries are reviewed and the formal model of certificicateless hybrid signcryption algorithm. In Sect. 3, we propose a concrete scheme under the new security model. Security and performance analysis of our scheme is included in Sect. 4. Finally, we draw some concluding remarks in Sect. 5.

2 Preliminaries

In this section, we briefly review some fundamental backgrounds such as bilinear maps, complexity assumptions and definition of the algorithm model. The following four definitions are quoted from [13, 17–19].

Definition 1 Let G_1 and G_2 be two cyclic multiplicative groups of prime order p, and a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ satisfying the following properties:

1. Bilinearity: $\hat{e}(aP, bP) = \hat{e}(P, P)^{ab}$ for any $a, b \in Z_q^*$.
2. Non-degeneracy: $\hat{e}(P, P) \neq 1_{G2}$.
3. Computability: $\hat{e}(P, Q)$ is efficiently computable for all $P, Q \in G_1$.

Definition 2 The challenger randomly chooses $a \in Z_q^*$ at random and given $(P, aP) \in G_1$. The computational elliptic curve discrete logarithm problem (ECDLP) is to compute a.

An adversary, $\mathcal{A}$, has a probability of at least ε in solving the ECDLP problem if

$$\Pr[\mathcal{A}(P, aP) = a] \geq \varepsilon$$

The ECDLP assumption holds if the advantage of any PPT adversary $\mathcal{A}$ is negligible in solving the ECDLP problem.

Definition 3 The challenger chooses $a, b \in Z_q^*$ at random and output (P, aP, bP). The computational Diffie–Hellman (CDH) problem is to compute abP.

An adversary, $\mathcal{A}$, has a probability of at least ε in solving the CDH problem if

$$\Pr[\mathcal{A}(P, aP, bP) = abP] \geq \varepsilon$$

The CDH assumption holds if the advantage of any PPT adversary $\mathcal{A}$ is negligible in solving the CDH problem.

Definition 4 The challenger randomly chooses $a, b, c, \mathrm{T} \in Z_q^*$ and sets $\delta \in \{0, 1\}$, if $\delta = 1$, and a 5-tuple $(P, aP, bP, cP, e(P, P)^{abc})$ is output, otherwise $(P, aP, bP, cP, e(P, P)^{T})$. The decisional Bilinear Diffie–Hellman (BDH) problem is to determine the value of T. An adversary $\mathcal{A}$ has at least an ε advantage in solving the BDH problem if

$$\left|\Pr\left[\mathcal{A}(P, aP, bP, cP, e(P, P)^{abc}) = 1\right]\right| - \left|\Pr\left[\mathcal{A}(P, aP, bP, cP, e(P, P)^{T}) = 1\right]\right| \geq 2\varepsilon$$

where the probability is defined over the randomly chosen a, b, c, T and the random bits consumed by $\mathcal{A}$. The BDH assumption holds if the advantage of any PPT adversary $\mathcal{A}$ is negligible in solving the BDH problem.

2.1 Formal model of certificateless hybrid signcryption

The notion of a certificateless hybrid signcryption scheme was defined by Cramer and Shoup [13]. A generic certificateless hybrid signcryption scheme works as following.

2.1.1 Key-Encapsulation-Mechanism

- *Setup* This algorithm takes as input a security k and returns system parameters *params* and a randomly chooses master secret key *msk*. After the algorithm is performed the KGC publishes the system parameters *params* and keeps the master key *msk* secret.
- *Extract-Partial-Private-Key* This algorithm takes as input *params*, *msk* and an identity $ID \in \{0,1\}^n$ of an entity, and returns a partial private key D_{ID}. The KGC carries out the algorithm after verifying the user's identity.
- *Generate-User-Key* This algorithm takes as input *params* and an identity *ID* and returns a randomly outputted secret value x_{ID} and a public key *PK*. This algorithm is run by a user to obtain a public key and a secret value which can be used to construct a full private key. Note that, the public key is published without certification.
- *Extract-Private-Key* This algorithm takes *params*, a user's partial private key D_{ID} and secret value x_{ID} as input, and returns the user's full private key *SK*. Obviously, the algorithm is executed by the entity itself.
- *Signcrypt* The algorithm takes *params*, a message m, a sender's identity *ID*, private key *SK* and public key *PK*, and a receiver's identity *ID* and public key *PK* as input, and returns a ciphertext or error symbol $\perp$.
- *Unsigncrypt* The algorithm takes a ciphertext, the receiver's identity *ID*, private key *SK* and public key *PK*, and a sender's identity *ID* and public key *PK* as input, and outputs a plaintext or an error symbol $\perp$.

2.1.2 Data-Encapsulation-Mechanism

- *Enc* this algorithm takes as input message $m \in \{0,1\}^n$ and encapsulation key K, then output a ciphertext $C \in 0,1^n$, where m is a bit string arbitrary length. We denote this as $C \leftarrow Enc_K(m)$.
- *Dec* this algorithm takes as input a key K and a ciphertext C, an outputs the message $m \in \{0,1\}^n$ or error symbol $\perp$.

Note that it is only required that the Data-Encapsulation-Mechanism is secure with respect to confidentiality and unforgeable.

2.2 Security Notions for Certificateless Hybrid Signcryption

Barbosa and Farshim [8] defined the formal security notions for CLSC scheme firstly, and scheme should satisfy confidentiality and unforgeability. As shown in [18–23], these security models are discussed in detail by considering two different type adversaries, Type I and II. A Type I adversary model an attacker which is a common user of the system and is not in possession of the KGC's master secret key, but it can replace the public key of an arbitrary entity, obtain the partial private key and private key. A Type II adversary model is an honest-but-curious KGC who knows the KGC's master secret key, but it cannot replace user's public key. While in the security game, the security model simulates the $game_A^{IND-CCA2}$ and $game_A^{EUF-CMA}$ between the adversary and challenger. And this paper gives the game simulation instance in Sect. 4.

3 Proposed Scheme

In this section, we illustrate a kind of practical certificateless hybrid signcryption scheme which consists of the following algorithm.

- *Setup* Given a security parameter k, the KGC performs the following to set up the system parameter.
  - Run the parameter generator on input k to generate a prime p, two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of same order $q\,(q > 2^k)$, a generator $P \in_R \mathbb{G}_1$ and an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
  - Selects the master secret key $s \in_R \mathbb{G}_q^*$ and the system public key is set to be $P_{pub} = sP$.
  - Selects three hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : (\{0,1\}^*)^2 \times (\mathbb{G}_1)^2 \times \mathbb{G}_2 \to \{0,1\}^n$ and $H_3 : \{0,1\}^l \times (\{0,1\}^*)^2 \times (\mathbb{G}_1)^4 \times \mathbb{G}_2 \to Z_q^*$.
  - Chooses a family of data encapsulation algorithm (*Enc*, *Dec*), and this algorithm is the confidentiality and unforgeable.
  - The public parameters of the scheme are set to be $params \leftarrow (q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3, Enc, Dec)$ and the master secret key is $msk \leftarrow s$.
- *Generate-User-Key* This algorithm takes as input *params* and a user's identity *ID*, it picks a random value $x_{ID} \in_R Z_q^*$ as user's secret value, and computes $PK_{ID} = x_{ID} \cdot P$ as the public key.
- *Extract-Partial-Private-Key* Given a group of user's identity $ID \in \{0,1\}^*$ and the corresponding public key PK_{ID}, the KGC generates a partial private key $D_{ID} = sH_1(ID \| PK_{ID})$ uniformly, and sends to user by using secure channel. Note that, the process of generate-partial-private-key is show in Fig. 1. Comparing to the traditional partial-private-key, getting partial-private-key of our scheme combined with the identity ID and public key PK_{ID}. Therefore, the partial-private-key can do some valid verification in the face of a threats to public key replacement attack.
- *Signcrypt* In order to signcrypt the message m to the receiver with identity ID_R and public key PK_R, and sender with public–private key pair $\{ID_S, PK_S, SK_S - (D_S, x_S)\}$ works as follow:
  - Choose $r_1, r_2 \in_R Z_q^*$, compute $R_1 = r_1 P, R_2 = r_2 P$, $Q_R = H_1(ID_R \| PK_R)$.
  - Compute $U = r_1 PK_R$, $V = \hat{e}(r_2 Q_R, P_{pub})$, $K = H_2(ID_S, ID_R, R_1, R_2, U, V)$, $\tau = Enc_K(m)$.
  - Compute $h = H_3(\tau, ID_S, ID_R, PK_S, PK_R, R_1, R_2, U, V)$, $W = h(D_S + r_2 Q_S)$, $T = hx_S + r_1$.

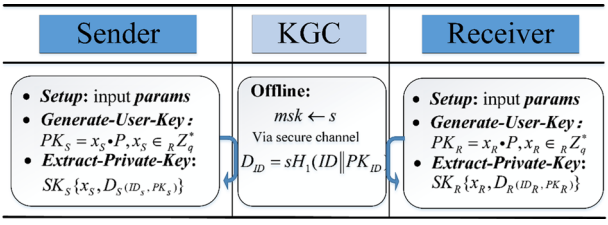

Fig. 1 The process of extract-partial-private-key

- Output $\sigma = (\tau, h, W, T)$ as the signcryption on m.

- *Unsigncrypt* To unsigncrypt a ciptertext $\sigma = (\tau, h, W, T)$ form the sender with identity ID_S and public key PK_S, the receiver with full public–private key $\{ ID_R, PK_R, SK_R - (D_R, x_R)\}$ works as follow:
  - Compute $Q_S = H_1(ID_S \| PK_S)$, $R_1 = TP - hPK_S$, $U' = x_R R_1'$, $V' = \hat{e}(D_R, R_2)$.
  - Check $h = H_3(\tau, ID_S, ID_R, PK_S, PK_R, R_1', R_2, U', V')$, $\hat{e}(W, P) = \hat{e}(hQ_S R_2 + P_{pub})$ hold or not.
  - If they hold, compute and output the decapsulation key $K' = H_2(ID_S, ID_R, R_1', R_2, U', V')$, else output symbol $''\perp''$.

3.1 Correctness

The correctness of our scheme is described as follow:

$$R_1' = TP - hPK_S = hx_S P + r_1 P - hPK_S = r_1 P = R_1$$

$$U' = x_R R_1' = r x_R P = r P_R = U$$

$$V' = \hat{e}(D_R, R_2) = \hat{e}(sQ_R, r_2 P) = \hat{e}(r_2 Q_R, sP) = \hat{e}(r_2 Q_R, P_{pub}) = V$$

Therefore, it is obvious that

$$h = H_3(\tau, ID_S, ID_R, PK_S, PK_R, R_1, R_2, U, V)$$

$$\begin{aligned} \hat{e}(W, P) &= \hat{e}(hD_S, P) \cdot \hat{e}(hr_2 Q_S, P) = \hat{e}(hQ_S, sP) \cdot \hat{e}(hQ_S, r_2 P) \\ &= \hat{e}(hQ_S, sP + r_2 P) = \hat{e}(hQ_S, R_2 + P_{pub}) \end{aligned}$$

Finally, we verify whether $K' = K$ holds or not, and recover the message m.

4 Proof of Security

Theorem 1 Under the CDH assumption and BDH assumption, and the data encapsulation algorithm (Enc, Dec) is confidentiality, we proposed scheme is Indistinguishability Against Adaptive Chosen Ciphertext Attacks (IND-CCA2) secure in the random oracle model. This theorem follows from Lemmas 1 and 2.

Lemma 1 Our scheme is IND-CCA2-I secure during the $game_{A_I}^{IND-CCA2-I}$, assuming that a probabilistic polynomial-time (PPT) adversary A_I (assume a dishonest user, but does not know system *msk*) has non-negligible advantage ε in winning this game against our scheme. More precisely, there exist an algorithm $\mathcal{C}$ which uses A_I to solve the BDH problem:

$$Adv_{scheme}^{IND-CCA2-I}(A_I) \geq \varepsilon \cdot \frac{1}{q_1}\left(1 - \frac{1}{q_2}\right) \cdot \left(1 - \frac{1}{q_{pa}}\right) \cdot \left(1 - \frac{q_S(2q_1 + q_2 + q_3)}{2^k}\right)\left(1 - \frac{q_{un}}{2^k}\right)$$

here q_i is denoted to query of hash oracle, q_{sk} extract private key query, q_{qk} extract public key query, q_{pa} request partial private key query, q_r public key replacement query, q_s signcryption query and q_{un} unsigncryption query.

Proof We suppose that the algorithm $\mathcal{C}$ is an example of BDH problem. $\mathcal{C}$ will run A_I as a subroutine and act as A_I's challenger in the $game_{A_I}^{IND-CCA2-I}$ for our scheme. By using this game, A_I will get various answers from each oracle and store these responses in the list which is empty at beginning. And the proof structure of random oracle model as shown in the Fig. 2.

Setup $\mathcal{C}$ runs setup algorithm and gives A_I the system $params(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3, Enc, Dec)$ with $P_{pub} \leftarrow aP$, where a as the system msk is unknown to $\mathcal{C}$. Then, $\mathcal{C}$ picks a challenged identity $J \in_R \{1, 2, \ldots, q_1\}$ randomly and answers oracle query as follows.

H_1-Query $\mathcal{C}$ maintains a hash list L_1^{list} of tuple $\langle ID_i, PK_i, Q_i, d_i \rangle$. When A_I ask the oracle H_1 at a point $\langle ID_j, PK_j, Q_j, d_j \rangle$, if the tuple already appears in the L_1^{list}, it returns the value. Otherwise, $\mathcal{C}$ picks a random $d_j \in_R Z_q^*$, computes $Q_j = d_j P$, and adds new tuple $\langle ID_j, PK_j, Q_j, d_j \rangle$ in the L_1^{list}. Note that, at the J-th query, $\mathcal{C}$ answers $Q_j = bP$ and puts the special tuple $\langle ID_J, PK_J, bP, \perp \rangle$ into the L_1^{list}.

H_2-Query $\mathcal{C}$ maintains a hash list L_2^{list} of tuple $\langle ID_i, ID_j, R_{1i}, R_{2i}, U_i, V_i, K_i \rangle$, when A_I query the oracle H_2, $\mathcal{C}$ checks the L_2^{list} and returns a unique value K_i if this tuple exists. Otherwise, $\mathcal{C}$ responses as follows:

1. If $\{j \neq J, U_i = x_j R_{1i} / x_j^* R_{1i}, V_i = e(d_j P, R_{2i})\}$ hold, $\mathcal{C}$ picks $K_i \in_R Z_q^*$ and returns this value, then adds into the L_2^{list}.
2. If $\{j = J, U_i = x_j R_{1i} / x_j^* R_{1i}, V_i = e(abP, R_{2i})\}$ hold, $\mathcal{C}$ picks $K_i \in_R Z_q^*$ and returns this value, then adds into the L_2^{list}

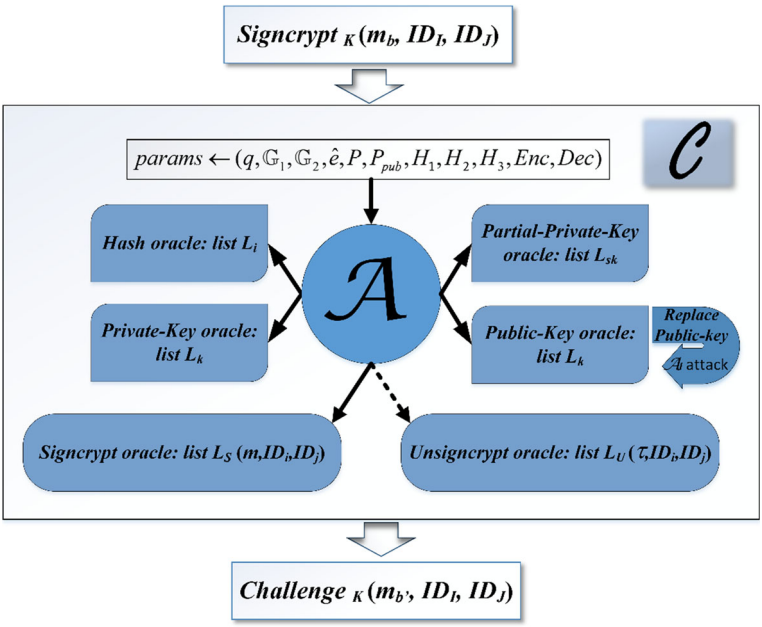

Fig. 2 The proof structure of random oracle model

3. Otherwise, $\mathcal{C}$ aborts and denotes the event by E1.

H_3-Query For each query $\langle \tau, ID_i, ID_j, PK_i, PK_j, R_{1i}, R_{2i}, U_i, V_i, h_i \rangle$, $\mathcal{C}$ checks the list L_3^{list} and returns h_i if these entities already exist in the L_3^{list}/L_S^{list}. Otherwise, $\mathcal{C}$ searches the L_1^{list} and L_2^{list}, picks a different $h_i \in_R Z_q^*$ to answer A_I, updates the L_3^{list} and the L_S^{list} at last.

Public-Key-Extract $\mathcal{C}$ maintains a public key list L_{PK}^{list} of tuple $\langle ID_i, PK_i, x_i \rangle$. For each query on PK_j, $\mathcal{C}$ will answer its value if the L_1^{list} or the L_{PK}^{list} contain this entity. Otherwise, $\mathcal{C}$ chooses $x_j \in_R Z_q^*$ randomly, generates a new key $PK_j = x_j P$ and returns this value, then updates the corresponding list.

Private-Key-Extract For each query on x_i, if the L_{PK}^{list} contain the corresponding entity, $\mathcal{C}$ adds it into the list L_{SK}^{list} of tuple $\langle ID_i, x_i, \perp \rangle$. If not, $\mathcal{C}$ makes a public key extraction query and updates this list.

Request-Partial-Private-Key When A_I ask a partial private key query, $\mathcal{C}$ will go through the list L_1^{list} and look for the corresponding index $\{ID_j, PK_j\}$. If $j \neq J$, $\mathcal{C}$ computes $D_j = d_j P_{pub}$, adds this tuple $\langle ID_i, x_i, D_j \rangle$ in the L_{SK}^{list}, and returns the value. If not, this game aborts and denotes the event by *E2*.

Public-Key-Replacement When A_I submits a identity ID_i and a certificateless public key PK_i^* to the L_{PK}^{list}, $\mathcal{C}$ inserts or updates the corresponding list with tuple $\langle ID_i, PK_i \leftarrow PK_i^*, \perp, \rangle$. Note that, A_I cannot query this private key pair from oracle.

Signcrypt-Query $\mathcal{C}$ maintains the L_S^{list} of tuple $\langle m, ID_i, ID_j, PK_i, PK_j, R_{1i}, R_{2i}, U_i, V_i, \tau_i, h_i, W_i, T_i \rangle$, where $\{ID_i, ID_j\}$ are the identity of sender and that of the receiver respectively. For each signcryption query, if $\{i \neq J, d_i \neq \perp, x_i/x_i^* \neq \perp\}$, where x_i^* input by A_I, $\mathcal{C}$ answers this query and follow with the signcrypt algorithm as Sect. 3, then generates the signcrytion ciphertext $\sigma \leftarrow (h_i, R_{2i}, W_i, T_i)$ as query result.

Note that, this game aborts and denotes the event by E3, if in this case $h_{L_3} \neq h_{L_S}$, where $\{h_{L_3}, h_{L_S}\}$ are the entity from the list L_3^{list} and L_S^{list} respectively.

Note also that, if $PK_i \leftarrow PK_i^*$, A_I inputs the value x_i^* and $\mathcal{C}$ computes $T_i = h_i x_i^* + r_1$, $W_i = h_i r_i Q_i^*$. If not, $\mathcal{C}$ computes $T_i = h_i x_i + r_1$, $W_i = h_i r_i Q_i$ and processes as usual.

Unsigncrypt-Query For each unsigncryption query $\langle m, ID_i, ID_j, PK_i, PK_j, R_{1i}, R_{2i}, U_i, V_i, \sigma \rangle$, where $\{ID_i, ID_j\}$ are the identity of sender and that of the receiver respectively, $\mathcal{C}$ proceeds as follows. Firstly, it obtains the receiver's partial private key $D_j = d_j P_{pub}$ and the sender's identity hash value Q_i corresponding to $\{ID_i, PK_i\}$ from the list L_1^{list}. Then, it executes the verification part of the unsigncryption algorithm as Sect. 3, returns $\perp$ if the verification does not succeed. After that, if $ID_j = ID_J$, $\mathcal{C}$ fails and stop (denote event by E4). Otherwise, it go through list and looks for a different entity $K^* \leftarrow (ID_i, ID_j, R_{1i}, R_{2i}, U_i, V_i)$. If such an entity exists, $\mathcal{C}$ returns this result $m'/\perp \leftarrow Dec_{K^*}(\sigma)$.

Challenge phase $\mathcal{C}$ generates two equal length plaintext (m_0, m_1) newly. And A_I picks two identities ID_S^* and ID_R^* on which it wishes to be challenged. If $ID_R^* \neq ID_J$, $\mathcal{C}$ fails and stops (denote event by E5). Otherwise, it proceeds to construct a challenge as follows. It randomly chooses $m_b (b \in_R \{0, 1\})$ and $r_1^* \in_R Z_q^*$, computes $R_1^* = r_1^* P$, $U^* = r_1^* P K_R$. Then it sets $R_2^* = cP$, computes $V^* = \hat{e}(cQ_j, P_{pub}) = \hat{e}(P, P)^{adc}$. After that, it computes $\tau^* =$

$Enc_{K^*}(m_b)$, $T^* = h^* x_S^* + r_1^*$, $W^* = h^*(D_S + cQ_S)$, where K^* is obtained from the L_2^{list}, h^* is obtained from the L_3^{list}. At last, $\mathcal{C}$ sends the challenge ciphertext $\sigma^* \leftarrow (h^*, R_2^*, W^*, T^*)$.

Guess phase A_I then performs a second series of queries which is treated in the same way as the first one (beyond the *Partial-Private-Key* query). At the end of the simulation, it produces the guess of challenge ciphertext m_b' for which it believes the $h^* \leftarrow (\tau^*, ID_S^*, ID_R^*, PK_S^*, PK_R^*, R_1^*, R_2^*, U^*, V^*)$, $e(W^*, P) = e(h^* Q_S, R_2^* + P_{pub})$, $\sigma^* \leftarrow Enc_{K^*}(m_b')$ hold. Therefore, it wins this game if $b' = b$ and game does not abortion. A_I has non-negligible advantage in winning the $game_{A_I}^{IND-CCA2-I}$ against our scheme, there exist an algorithm $\mathcal{C}$ which use A_I to solve the BDH problem such that:

$$Adv_{scheme}^{IND-CCA2-I}(A_I) = \varepsilon \cdot \Pr[\neg E1 \wedge \neg E2 \wedge \neg E3 \wedge \neg E4 \wedge \neg E5]$$
$$\geq \varepsilon \cdot \frac{1}{q_1}\left(1 - \frac{1}{q_2}\right) \cdot \left(1 - \frac{1}{q_{pa}}\right) \cdot \left(1 - \frac{q_S(2q_1 + q_2 + q_3)}{2^k}\right)\left(1 - \frac{q_{un}}{2^k}\right)$$

Lemma 2 Our scheme is IND-CCA2-II secure during the $game_{A_{II}}^{IND-CCA2-II}$, assuming that a PPT adversary $\mathbb{A}_{II}$ (assume an honest but curious KGC, and it cannot replace user's key) has non-negligible advantage ε in winning this game against our scheme. More precisely, there exist an algorithm $\mathcal{C}$ which uses $\mathbb{A}_{II}$ to solve the CDH problem:

$$Adv_{scheme}^{IND-CCA2-II}(A_{II}) \geq \varepsilon \cdot \frac{1}{q_1} \cdot \left(1 - \frac{1}{q_2}\right) \cdot \left(1 - \frac{1}{q_{sk}}\right) \cdot \left(1 - \frac{q_S(2q_1 + q_2 + q_3)}{2^k}\right)\left(1 - \frac{q_{un}}{2^k}\right)$$

here $q_1, q_2, q_3, q_{pk}, q_{sk}, q_{pa}, q_S$ and q_{un} denote the same as in Lemma 1.

Proof The algorithm $\mathcal{C}$, solving example of CDH problem, simulation process is similar to Lemma 1. But the difference is that $\mathbb{A}_{II}$ will have different way of answer.

Setup $\mathcal{C}$ runs setup algorithm and gets the system $params(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3, Enc, Dec)$, then $\mathcal{C}$ sends both (*params, msk*) to $\mathbb{A}_{II}$.

H_1-Query For each H_1 query, $\mathcal{C}$ goes through the list L_1^{list} if containing a tuple $\langle ID_i, PK_i, Q_i, d_i\rangle$, it returns the value. Otherwise, $\mathcal{C}$ chooses a random value $d_i \in Z_q^*$ and puts the new entries $\langle ID_i, PK_i, Q_i, d_i\rangle$ into the list L_1^{list}, answers $Q_i = d_i P$ in the end.

H_2-Query same as the Lemma 1.

H_3-Query H_3same as the Lemma 1.

Public-Key-Extract For each query on PK_i, if $j \neq J$, $\mathcal{C}$ generates a usually key pair $\{PK_i, x_i\}$ and updates these corresponding list. Otherwise, $\mathcal{C}$ returns $PK_j \leftarrow aP$, adds into the L_1^{list} of tuple $\langle ID_j, aP, Q_j, d_j\rangle$ and the L_{PK}^{list} of tuple $\langle ID_i, \perp, aP\rangle$.

Private-Key-Extract For each query on x_i, if the L_{PK}^{list} contain the corresponding entity, $\mathcal{C}$ adds it into the list L_{SK}^{list} of tuple $\langle ID_i, x_i, \perp\rangle$. If not, $\mathcal{C}$ makes a public key extraction query or fails this game at J-th query (denote event by E2).

Request-Partial-Private-Key When $\mathbb{A}_{II}$ ask a partial private key query on identity ID_j, $\mathcal{C}$ goes through the L_1^{list} and looks for these corresponding entities $\{ID_j, PK_j\}$, computes $Dj = d_j P_{pub}$, then adds tuple $\langle ID_j, x_j, D_j\rangle$ into the L_{SK}^{list}.

Signcrypt-Query $\mathcal{C}$ maintains the L_S^{list} of tuple $\langle m, ID_i, ID_j, PK_i, PK_j, R_{1i}, R_{2i}, U_i, V_i, \tau_i, h_i, W_i, T_i\rangle$, where $\{ID_i, ID_j\}$ are the identity of sender and that of the receiver respectively. For each signcryption query, if $\{i \neq J, x_i \neq \perp\}$, $\mathcal{C}$ answers this query and follow with the signcrypt algorithm as Sect. 3, then generates the signcrytion ciphertext $\sigma \leftarrow (h_i, R_{2i}, W_i, T_i)$ as query result. If $\{i = J, x_i = \perp\}$, $\mathcal{C}$ chooses $\{r_2, h, T\} \in_R Z_q^*$ randomly, and computes $R_1 = TP - h_i PK_i$, $R_{2i} = r_2 P$, $U = x_j R$, $V = \hat{e}(r_2 Q_j, P_{pub})$, then returns the result $\sigma \leftarrow (h_i, R_{2i}, W_i, T_i)$.

Note that, this game fails and stops (denotes the event by E3), if $h_{L_3} \neq h_{L_S}$, where $\{h_{L_3}, h_{L_S}\}$ are the entity from the list L_3^{list} and L_S^{list} respectively.

Unsigncrypt-Query For each unsigncryption query $\langle m, ID_i, ID_j, PK_i, PK_j, R_{1i}, R_{2i}, U_i, V_i, \sigma\rangle$, where $\{ID_i, ID_j\}$ are the identity of sender and that of the receiver respectively, $\mathcal{C}$ works as follows. Firstly, it obtains the receiver's partial private key $D_j = d_j P_{pub}$ and the sender's identity hash value Q_i corresponding to $\{ID_i, PK_i\}$ from the list L_1^{list}. Then, it executes the verification part of the unsigncryption algorithm as subsection 3.2, returns $\perp$ if the verification does not succeed. After that, if $ID_j = ID_J$, $\mathcal{C}$ fails and stop (denote event by E4). Otherwise, it go through list and looks for a different entity $K^* \leftarrow (ID_i, ID_j, R_{1i}, R_{2i}, U_i, V_i)$. If such an entity exists, $\mathcal{C}$ returns this result $m'/\perp \leftarrow Dec_{K^*}(\sigma)$.

Challenge phase $\mathcal{C}$ generates two equal length plaintext (m_0, m_1) newly. And $\mathbb{A}_{II}$ picks two identities ID_S^* and ID_R^* on which it wishes to be challenged. If $ID_R^* \neq ID_J$, $\mathcal{C}$ fails and stops (denote event by E5). Otherwise, it proceeds to construct a challenge as follows. It randomly chooses $m_b (b \in_R \{0, 1\})$ and $r_2^* \in_R Z_q^*$, computes $R_1^* = bP$, $U^* = abP$, $V^* = \hat{e}(r_2 Q_J, P_{pub})$. After that, it computes $\tau^* = Enc_{K^*}(m_b)$, $T^* = h^* x_S^* + r_1^*$, $W^* = h^*(D_S + cQ_S)$, At last, $\mathcal{C}$ returns $\sigma^* \leftarrow (h^*, R_2^*, W^*, T^*)$ to $\mathbb{A}_{II}$ as the challenge ciphertext.

Guess phase $\mathbb{A}_{II}$ then performs a second series of queries which is treated in the same way as the first one (prohibit the ***Private-Key*** query). At the end of the simulation, it produces the guess of challenge ciphertext m_b' for which it believes the $h^* \leftarrow (\tau^*, ID_S^*, ID_R^*, PK_S^*, PK_R^*, R_1^*, R_2^*, U^*, V^*)$, $e(W^*, P) = e(h^* Q_S, R_2^* + P_{pub})$, $\sigma^* \leftarrow Enc_{K^*}(m_b')$ hold. Therefore, it wins this game if $b' = b$ and game does not abortion. More precisely, $\mathbb{A}_{II}$ have non-negligible advantage in winning the $game_{A_{II}}^{IND-CCA2-II}$ against our scheme, there exist an algorithm $\mathcal{C}$ which use $\mathbb{A}_{II}$ to solve the CDH problem such that:

$$
\begin{aligned}
Adv_{scheme}^{IND-CCA2-II}(A_{II}) &= \varepsilon \cdot \Pr[\neg E1 \wedge \neg E2 \wedge \neg E3 \wedge \neg E4 \wedge \neg E5] \\
&\geq \varepsilon \cdot \frac{1}{q_1} \cdot \left(1 - \frac{1}{q_2}\right) \cdot \left(1 - \frac{1}{q_{sk}}\right) \cdot \left(1 - \frac{q_S(2q_1 + q_2 + q_3)}{2^k}\right)\left(1 - \frac{q_{un}}{2^k}\right)
\end{aligned}
$$

Theorem 2 Under the CDH assumption and ECDL assumption, and the data encapsulation algorithm (Enc, Dec) is confidentiality, we proposed scheme is Existentially Unforgeable Against Chosen Message Attacks (EUF-CMA) secure in the random oracle model. This theorem follows from Lemmas 3 and 4.

Lemma 3 We proposed scheme is EUF-CMA -I secure during the $game_{A_I}^{EUF-CMA-I}$, assuming that a PPT adversary A_I has non-negligible advantage ε in winning this game against our scheme. More precisely, there exist an algorithm $\mathcal{C}$ which uses A_I to solve the CDH problem with probability:

$$Adv_{scheme}^{EUF-CMA-I}(A_I) \geq \varepsilon \cdot \frac{1}{q_1}\left(1-\frac{1}{q_2}\right)\cdot\left(1-\frac{1}{q_{pa}}\right)\cdot\left(1-\frac{q_S(2q_1+q_2+q_3)}{2^k}\right)$$

here $q_1, q_2, q_3, q_{pk}, q_{sk}, q_{pa}, q_S$ and q_{un} denote the same as in Lemma 1.

Proof In query phase, A_I can ask a polynomially bounded number of queries adaptively again as Lemma 1 (prohibit the signcrypt-query). At the end of the simulation, A_I outputs a group of challenge ciphertex $\sigma^* \leftarrow (h^*, R_2^*, W^*, T^*)$, and $\mathcal{C}$ checks $ID_S^* \neq ID_J$. If not, it aborts this game. Otherwise, it obtains r_2^* and h^* by calling the hash oracle and retrieves Q_S from the list L_1^{list} to computing the answer of CDH problem:

$$\begin{aligned}
\hat{e}(W^*, P) &= \hat{e}(h^*Q_S, R_2^* + P_{pub}) = \hat{e}(h^*Q_S, P_{pub})\hat{e}(h^*Q_S, R_2^*)\\
\hat{e}(W^*, P) &= \hat{e}(bP, aP)^{h^*}\hat{e}(h^*bP, r_2^*P)\\
\hat{e}(abP, P) &= \hat{e}\left(\frac{W^*}{h^*} - r_2^*Q_S, P\right)
\end{aligned}$$

Thus $\mathcal{C}$ can recover $abP = \frac{W^*}{h^*} - r_2^*Q_S$ as the return of the CDH problem. And A_I has non-negligible advantage ε in winning the $game_{A_I}^{EUF-CMA-I}$ against our scheme, an algorithm $\mathcal{C}$ to solve the CDH problem with probability:

$$Adv_{scheme}^{EUF-CMA-I}(A_I) \geq \varepsilon \cdot \frac{1}{q_1}\left(1-\frac{1}{q_2}\right)\cdot\left(1-\frac{1}{q_{pa}}\right)\cdot\left(1-\frac{q_S(2q_1+q_2+q_3)}{2^k}\right)$$

Lemma 4 We proposed scheme is EUF-CMA -II secure during the $game_{A_{II}}^{EUF-CMA-II}$, assuming that a PPT adversary $\mathbb{A}_{II}$ has non-negligible advantage ε in winning this game against our scheme. More precisely, there exist an algorithm an algorithm $\mathcal{C}$ which uses $\mathbb{A}_{II}$ to solve the ECDL problem with probability:

$$Adv_{scheme}^{IND-CCA2-II}(A_{II}) \geq \varepsilon \cdot \frac{1}{q_1}\cdot\left(1-\frac{1}{q_2}\right)\cdot\left(1-\frac{1}{q_{sk}}\right)\cdot\left(1-\frac{q_S(2q_1+q_2+q_3)}{2^k}\right)$$

here $q_1, q_2, q_3, q_{pk}, q_{sk}, q_{pa}, q_S$ and q_{un} denote the same as in Lemma 1.

Proof The proof program is same as the Lemmas 2 and 3.

5 Efficiency of Our Scheme

Since computational overhead and ciphertext size are two important factors affecting the efficiency, we present the compassion with the existing CLSC schemes in this section [10, 12, 17, 18]. In view of the computation cost, we focus on the costly operations and omit the computation efforts which can be pre-computed. Note that, the pairing operations is several times more expensive than other operation [24, 25]. And we use *Mult, Exp* and *Pair* as abbreviations for point multiplications, exponentiations and pairing computations respectively. We also use $|G|, |r|$ and $|m|$ to denote the size of an element in G_1, and the size of an element in finite field Z_q^* and the length if message m. From Table 1, we can observe that communication overhead and computational cost of our scheme are more efficient than the relating schemes.

Table 1 efficiency comparison with other schemes

Schemes	Ciphertext size	Signcryption			Unsigncryption		
		Mult	*Exp*	*Pair*	*Mult*	*Exp*	*Pair*
Ref. [10] 2008	$2\|G\| + \|m\|$	4	4	1	1	4	3
Ref. [12] 2010	$2\|G\| + \|r\| + \|m\|$	3	3	2	0	1	3
Ref. [17] 2013	$2\|G\| + \|m\|$	4	1	1	1	0	5
Ref. [18] 2014	$2\|G\| + \|m\|$	4	1	1	0	1	5
Ours	$\|G\| + 2\|r\| + \|m\|$	3	0	1	3	0	3

6 Conclusion

In this paper, we have discussed a practical certificateless hybrid signcryption scheme, which not only can signcrypt arbitrary length data, but also can guarantee scheme efficiently against the public-key-replacement attacks by using the technique of the public key binding in the extract-partial-private-key. Furthermore, this scheme has been proved its confidentiality and unforgeable in the random oracle model. According to the comparison with other related schemes, the new scheme is efficient and practical.

Acknowledgments This work was partially supported by the National Science Foundation of China under Grants 61262079.

References

1. Uhsadel, L., Ullrich, M., Das, A., et al. (2013). Teaching HW/SW co-design with a public key cryptography application. *IEEE Transactions on Education, 56*(4), 478–483.
2. Chan, S., Guizani, M., Chen, C., et al. (2014). An enhanced public key infrastructure to secure smart grid wireless communication networks. *IEEE Network*, *28*(1), 10–16.
3. Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *CRYPTO'84. Lecture notes in computer science* (Vol. 196, pp. 47–53). Springer: Heidelberg.
4. Malone-Lee, J. (2002). Identity-based signcryption. *IACR Cryptology ePrint Archive, 2002*, 98.
5. Hölbl, M., Welzer, T., & Brumen, B. (2012). An improved two-party identity-based authenticated key agreement protocol using pairings. *Journal of Computer and System Sciences, 78*(1), 142–150.
6. Yin, A., Liang, H., & Zhu, M. (2014). Authentication protocol using MYK-NTRUSign signature algorithm in wireless network environment. *Journal of Networks, 9*(5), 1139–1144.
7. Al-Riyami, S. S., & Paterson, K. G. (2003). Certificateless public key cryptography. In *Proceedings of ASIACRYPT 2003. Lecture notes in computer science* (Vol. 2894, pp. 452–473). Heidelberg: Springer.
8. Barbosa, M., & Farshim, P. (2008). Certificateless signcryption. In *Proceedings of ASIACCS'2008* (pp. 369–372). New York: ACM.
9. Han, Y. L., & Gui, X. L. (2009). BPGSC: Bilinear pairing based generalized signcryption scheme. In *2009 eighth international conference on grid and cooperative computing* (pp. 76–82) Lanzhou.
10. Wu, C., & Chen, Z. (2008). A new efficient certificateless signcryption scheme. In *International symposium on information science and engineering, 2008 (ISISE'08)* (Vol. 1, pp. 661–664). Shanghai: IEEE.
11. Selvi, S. S. D., Vivek, S. S., & Rangan, C. P. (2009). On the security of certificateless signcryption schemes. *INSCRYPT, 9*, 75–92.
12. Selvi, S. S. D., Vivek, S. S., & Rangan, C. P. (Eds.). (2011). Cryptanalysis of certificateless signcryption schemes and an efficient construction without pairing. In *Information security and cryptology* (pp. 75–92). Berlin, Heidelberg: Springer.
13. Cramer, R., & Shoup, V. (2003). Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing, 33*(1), 167–226.

14. Kurosawa, K., & Desmedt, Y. (2004). A new paradigm of hybrid encryption scheme. *Lecture Notes in Computer Science, 3152*, 426–442.
15. Chen, L., Cheng, Z., Malone-Lee, J., & Smart, N. P. (2006). Efficient ID-KEM based on the Sakai-Kasahara key construction. *IEE Proceedings-Information Security, 153*, 19–26.
16. Bentahar, K., Farshim, P., Malone-Lee, J., & Smart, N. P. (2008). Generic constructions of identity-based and certificateless KEMs. *Journal of Cryptology, 21*, 178–199.
17. Li, F., Shirase, M., & Takagi, T. (2013). Certificateless hybrid signcryption. *Mathematical and Computer Modelling, 57*(3), 324–343.
18. Weng, J., Yao, G. X., Deng, R. H., et al. (2011). Cryptanalysis of a certificateless signcryption scheme in the standardmodel. *Information Sciences, 181*, 661–667.
19. Selvi, S. S. D., Vivek, S. S., & Rangan, C. P. (2010). Security weaknesses in two certificateless signcryption schemes. *IACR Cryptology ePrint Archive, 2010*, 92–95.
20. Zhou, C., Zhou, W., & Dong, X. (2014). Provable certificateless generalized signcryption scheme. *Designs, Codes and Cryptography, 71*(2), 331–346.
21. Liu, Z., Hu, Y., Zhang, X., et al. (2010). Certificateless signcryption scheme in the standard model. *Information Sciences, 180*(3), 452–464.
22. Boneh, D., & Boyen, X. (2011). Efficient selective identity-based encryption without random oracles. *Journal of Cryptology, 24*(4), 659–693.
23. Herranz, J., Ruiz, A., & Sáez, G. (2014). Signcryption schemes with threshold unsigncryption, and applications. *Designs, Codes and Cryptography, 70*(3), 323–345.
24. Galbraith, S. D., Paterson, K. G., & Smart, N. P. (2008). Pairings for cryptographers. *Discrete Applied Mathematics, 156*(16), 3113–3121.
25. Granger, R., & Smart, N. P. (2006) On computing products of pairings. *IACR Cryptology ePrint Archive, 2006*, 172–183.

Aihan Yin is a Professor of Information Engineering department at East China Jiaotong University in Jiangxi, China, since 2003. She received the Bachelor's degree from TIANJIN University in 1984, and the Master's degree from the Nanjing University of Aeronautics and Astronautics in 2005, respectively. In 2011, she obtained her Ph.D. at Huazhong University of Science and Technology. Her research focuses on Optic Communication Technology, Communication Network Protocol, Wireless Communication Technology, Signal Processing and Optoelectronic Technique.

Hongchao Liang is a postgraduate student at East China Jiaotong University in Jiangxi, China. He received his bachelor in Communication engineering department at East China Jiaotong University in Jiangxi, China. His interests include authentication and encryption technology in the wireless network.