Analysis of Certificateless Signcryption Schemes and Construction of a Secure and Efficient Pairing-free one based on ECC

Liling Cao[1*] and Wancheng Ge[2]
[1] Department of Engineering Science and Technology, Shanghai Ocean University
Shanghai, 201306 - China
[e-mail: llcao@shou.edu.cn;]
[2] Department of Electronic and Information Engineering, Tongji University
Shanghai, 201804 - China
[e-mail: gwc828@tongji.edu.cn]
*Corresponding author: Liling Cao

Received May 7, 2016; revised December 14, 2017; accepted April 6, 2018;
published September 30, 2018

Abstract

Signcryption is a cryptographic primitive that provides authentication (signing) and confidentiality (encrypting) simultaneously at a lower computational cost and communication overhead. With the proposition of certificateless public key cryptography (CLPKC), certificateless signcryption (CLSC) scheme has gradually become a research hotspot and attracted extensive attentions. However, many of previous CLSC schemes are constructed based on time-consuming pairing operation, which is impractical for mobile devices with limited computation ability and battery capacity. Although researchers have proposed pairing-free CLSC schemes to solve the issue of efficiency, many of them are in fact still insecure. Therefore, the challenging problem is to keep the balance between efficiency and security in CLSC schemes. In this paper, several existing CLSC schemes are cryptanalyzed and a new CLSC scheme without pairing based on elliptic curve cryptosystem (ECC) is presented. The proposed CLSC scheme is provably secure against indistinguishability under adaptive chosen-ciphertext attack (IND-CCA2) and existential unforgeability under adaptive chosen-message attack (EUF-CMA) resting on Gap Diffie-Hellman (GDH) assumption and discrete logarithm problem in the random oracle model. Furthermore, the proposed scheme resists the ephemeral secret leakage (ESL) attack, public key replacement (PKR) attack, malicious but passive KGC (MPK) attack, and presents efficient computational overhead compared with the existing related CLSC schemes.

Keywords: certificateless; ECC; pairing-free; signcryption; random oracle

This research was supported by the Doctoral Scientific Research Founation of Shanghai Ocean University (A2-0203-00-100361), "innovative action plan" of Science and Technology Commission of Shanghai Municipality (17050502000). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

http://doi.org/10.3837/tiis.2018.09.022 **ISSN : 1976-7277**

1. Introduction

Traditional public key infrastructure (TPKI) cryptosystem[1], which suffers from complicated public key certificate management, is impractical for mobile devices with limited computation ability and battery capacity. An effective substitution for traditional PKI cryptosystem without the operation of certificate is identity-based (ID-based) cryptosystem initially proposed by Shamir[2], in which the public key of the user is easily computed from the identity of the user such as IP address or email address, while the private key is generated from the identity of the user and a master secret key of a key generator center (KGC) known as a trusted authority. To reduce the heavy trust reliance on KGC, in 2003, Al-Riyami and Paterson[3] presented a novel concept called certificateless public key cryptography (CLPKC), in which long-term private key of the user is calculated from a secret key of the user, while partial private key of the user is issued by KGC. In this way, CLPKC-based protocols eliminate the complex certificate management burden and the insecure key escrow problem, which respectively consists in TPKI and ID-based cryptosystems.

In information and network applications, encryption technique and digital signature are two fundamental mechanisms explored to match specific security requirements, including confidentiality, integrity, non-repudiation and authentication. Traditionally, signing and encrypting the message are independent with an encrypt-then-sign paradigm. Signcryption, put forward by Zheng [4] in 1997, is a cryptographic primitive that provides authentication (signing) and confidentiality (encrypting) simultaneously, at a lower computational cost and communication overhead. Previously, researchers constructed signcryption schemes based on TPKI and ID-based cryptosystems. Recently, the explosive growth of security and performance requirements has necessitated extensive researches on certificateless signcryption (CLSC) schemes owing to the satisfactory performances of CLPKC.

1.1. Related studies

Certificateless signcryption (CLSC) schemes can be divided into two categories according to the way of computing in the schemes, (1) Pairing-based CLSC schemes, (2) Pairing-free CLSC schemes.

CLSC scheme was firstly put forward by Barbosa and Farshim [5] in 2008 and previous CLSC schemes were relying on costly bilinear pairing operations. In traditional pairing-based CLSC schemes, a particular collection of a message part is required to be signcrypted and sent. That means a large message should be divided into several sections, each of which should match the size of input for signcryption in *signcrypt* algorithm. Hereafter, many traitional CLSC schemes [6-13] relying on pairing operations have been proposed. In 2008, Aranha et al. [6] and Wu et al. [7] proposed two schemes separately. In 2010, Liu et al. [9] figured out that Barbosa and Farshim's CLSC scheme [5] was insecure under malicious but passive KGC (MPK) attack [14] and constructed an improved one, which was unfortunately proved to be insecure against MPK attack either, as indicated by Weng et al. [15]. That same year, Selvi et al. also demonstrated security weaknesses of the schemes [5-7] in their literature [16]. Compared with CLSC schemes [7] and [9], Xie et al. [8] improved a more efficient one, which was, however, vulnerable to ephemeral secret leakage (ESL) attack [17], as analyzed by Hafizul Islam et al.[10], who then proposed a leakage-free CLSC scheme with security against ESL attack in the random oracle model in 2015.

Besides, there were some untraditional pairing-based CLSC schemes proposed by reseachers. In 2013, Li et al. [11] generated a novel hybrid CLSC scheme, in which a message should not be divided into appropriate sections. In such a construction, a symmetric key, which will be used to encrypt the actual message later, is signcrypted and sent from the signer. It is worthwhile to analyze such different paradigm from traditional research works, because a ***signcrypt*** algorithm is also adopted in their schemes. A symmetric key is signcrypted in such special schemes, while a section of the message is signcryped in traditional ones. In addition, in 2014, Zhou et al. [12] introduced a provable certificateless generalized signcryption scheme, which could adaptively work as an encryption scheme, a signature scheme or a signcryption scheme with only one algorithm. Such algorithms running in signcryption mode, which are equivalent to the traditional CLSC schemes, are also worth discussing in our research works.With studies on the two schemes above, Yin et al. [13] demonstrated that these two schemes were inefficient with higher computation cost compared with their own proposed improved scheme.

Nevertheless, all CLSC schemes mentioned above are relying on costly bilinear pairing operations, which are impractical for mobile devices with limited computation ability and battery capacity.

Therefore, it is significant and challenging to come up with secure and efficient pairing-free CLSC schemes, which provide more security properties without complicated operations. In 2010, Selvi et al.[16] presented the first provably secure CLSC scheme without bilinear pairing and validated it in the random oracle model. Among the existing pairing-free CLSC schemes [18-20], He [21] claimed that scheme [19] failed to achieve unforgeability property when the Type I adversary executed attacks. In 2014, Shi et al. [22] claimed that all the CLSC schemes in [18-20] provided neither unforgeability nor confidentiality property against the Type I adversary. Moreover, in 2014, Lu et al. [23] proposed a certificate-based signcryption scheme without costly bilinear operations. The ceriticate produced by the ***Certify*** algorithm in their scheme is equivalent to the partial private key produced by the ***Extract Partial Private Key*** algorithm in traditional CLSC schemes. Lu et al. claimed that the ceriticates could be sent to the users publicly, which resolved the distribution problem in CLPKC. However, security model in their scheme includes a Type I adversary who has no access to the certificates, which is contradictory to the ***Certify*** algorithm. In fact, Lu et al.'s scheme is an implicit CLSC scheme.

Some recent research works on pairing-based and pairing-free CLSC schemes are summarized in **Fig. 1** and **Fig.2** respectively. Researchers at the end of the arrow indicated that the schemes proposed by the researchers at the beginning of the arrow was either insecure or incorrect. Besides, there was no impoved CLSC schemes presented in [15] and [21].

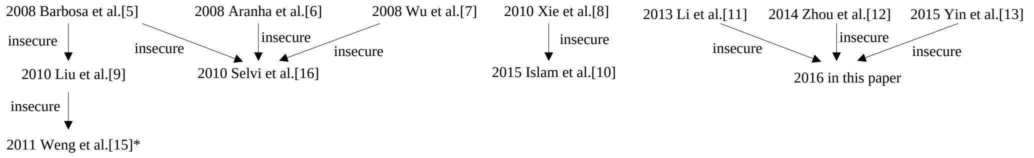

Fig. 1. Recent research woks on pairing-based CLSC schemes

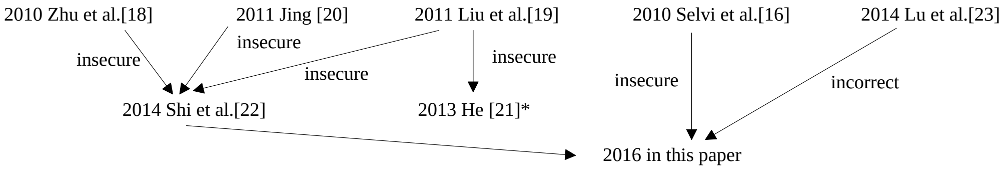

Fig. 2. Recent research woks on pairing-free CLSC schemes

1.2. Our contribitions

In this paper, we analyze schemes [11-13, 16, 22] by concrete cryptanalysis. All these schemes are vulnerable to ephemeral secret leakage (ESL) attack, public key replacement (PKR) attack, malicious but passive KGC (MPK) attack, and not secure enough to provide confidentiality or unforgeability property. Motivated by the prior research works, we construct a secure and efficient pairing-free CLSC scheme based on ECC. Compared with existing CLSC schemes, our proposed scheme achieves greater security with lower computation cost.

2. Preliminaries

2.1. Security assumption based on ECC

Let F_p be a finite prime field with a large prime number p. An elliptic curve E over the finite field F_p is the set of all pairs satisfying the equation $y^2(mod\ p) = x^3 + ax + b\ (mod\ p)$, $a, b \in Fp, \Delta = 4a^3 + 27b^2(mod\ \ p) \neq 0$, along with an imaginary point representing the infinity. An additive group G_p of all points on elliptic curve E includes addition operation.

Let P be a generator of G_p. Let the order of G_p be an integer q. Let $Z_q^* = [1, q - 1]$. Following computational problems over the elliptic curve E are frequently used in cryptographic protocols. The probability to solve these problems is negligible with any polynomial time algorithm.

Discrete Logarithm (DL) problem: for unknown $a \in Z_q^*$, by giving $P, aP, P \in E/\,Fp$, compute a.

Computational Diffie-Hellman (CDH) problem: for unknown $a, b \in Z_q^*$, by giving $P, aP, bP, P \in E/\,Fp$, compute abP.

Decision Diffie-Hellman (DDH) problem: for unknown $a, b, c \in Z_q^*$, by giving $P, aP, bP, cP, P \in E/\,Fp$, decide whether $abP = cP$.

Gap Diffie-Hellman (GDH) problem: for unknown $a, b \in Z_q^*$, by giving $P, aP, bP, P \in E/\,Fp$ and an oracle DDH(aP, bP, cP), that outputs 1 if $abP = cP$, otherwise 0, compute abP.

2.2. Structure of CLSC schemes

Notions used in this paper are listed in **Table 1**. CLSC scheme, which consists of seven polynomial time algorithms, can be summarized in **Table 2** according to the following expression.

$$\{outputs\} \xleftarrow{\text{algorithm executive}} \text{algorithm}(inputs)$$

Table 1. Notations used in this paper

Notation	Description
ID_i	the identity of participant i
H (*)	secure collision-free one-way hash functions
(s, P_{pub})	the KGC's master secret key/public key pair
(x_i, P_i)	secret value/public key pair of participant i, P_i is calculated from x_i
d_i	d_i is the partial private key of participant i
r_i	a random number generated by sender i (i.e. ephemeral private key) for signcryption
(sk_i, pk_i)	private key/public key pair of participant i, where sk_i=(x_i, d_i), pk_i=(P_i)
k	security parameter set by KGC
m/σ	message plaintext / ciphertext with k bits
$\perp$	represents no message or an unknown value

For example, $\{pk_i\} \xleftarrow{\text{user } i} \text{PUK}(ID_i, x_i, system\ params)$ means that user i executes PUK algorithm to generate public key pk_i by taking $ID_i, x_i, system\ params$ as inputs.

Table 2. Algorithms of a CLSC scheme

Algorithm Name(Abbreviation)	Expression
setup(SETUP)	$\{s, system\ params\} \xleftarrow{\text{KGC}} \text{SETUP}(k)$
Extract Partial Private Key (EPRK)	$\{d_i\} \xleftarrow{\text{KGC}} \text{EPRK}(s, ID_i, system\ params)$
Set Secret Value(SV)	$\{x_i\} \xleftarrow{\text{user } i} \text{SV}(ID_i, system\ params)$
Set Private Key(PRK)	$\{sk_i\} \xleftarrow{\text{user } i} \text{PRK}(d_i, x_i)$
Set Public Key(PUK)	$\{pk_i\} \xleftarrow{\text{user } i} \text{PUK}(ID_i, x_i, system\ params)$
Signcrypt (SC)	$\{\sigma\} \xleftarrow{\text{sender } i} \text{SC}(m, ID_i, sk_i, pk_i, ID_j, pk_j, system\ params)$
Unsigncrypt (USC)	$\{m\ or\ \perp\} \xleftarrow{\text{receiver } j} \text{USC}(\sigma, ID_i, pk_i, ID_j, sk_j, pk_j)$

2.3. Security model

2.3.1 Adversary model

There are two kinds of adversaries in CLPKC. $\mathcal{A}_1$, as a dishonest user, can replace the public key of any user with a value x_i of his choice, but cannot access the master secret key of KGC. $\mathcal{A}_2$, as a malicious but passive KGC, cannot replace the public keys, but can obtain the master secret key of KGC.

2.3.2 Security model

The security model is defined as an attack game between an adversary $\mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_2\}$ and a challenger $\mathcal{C}$ in a series of simulated potential attacking scenarios. The adversary, simulated as a user, asks the challenger for a polynomial number of queries, while the challenger issues the replies using the following oracles.

(1) Create(ID_i): $\mathcal{C}$ generates private key/public key pair (sk_i,pk_i) of participant i.

(2) Rd_i: $\mathcal{C}$ reveals to $\mathcal{A}$ the partial private key d_i of participant i.

(3) Rx_i: $\mathcal{C}$ reveals to $\mathcal{A}$ the secret value x_i of participant i.

(4) Rsk_i: $\mathcal{C}$ reveals to $\mathcal{A}$ the private key (d_i, x_i) of participant i.

(5) Rpk_i: $\mathcal{C}$ replaces the public key of participant i with the value x_i chosen by $\mathcal{A}$, which means that the secret values of all participants can be set by $\mathcal{A}$.

(6) Rr_i: $\mathcal{C}$ reveals to $\mathcal{A}$ the ephemeral private key r_i of participant i.

(7) R$_{sc}(m, ID_i, ID_j)$: With the identity of the sender ID_i, the identity of the receiver ID_j and the message m, $\mathcal{C}$ executes signcryption algorithm and outputs σ or $\perp$.

(8) R*usc* (σ, ID_i, ID_j): With the identity of the sender ID_i, the identity of the receiver ID_j and the ciphertext σ, Ϲ executes unsigncryption algorithm and outputs m or ⊥.

Definition 1 (Confidentiality): A CLSC scheme satisfies confidentiality against indistinguishability under adaptive chosen-ciphertext attack (IND-CCA2) only if the probability for attackers to win the following game is negligible with any polynomial time algorithm.

Steps of the **Game IND-CCA2** are described as follows.

(C1) The challenger Ϲ executes the SETUP algorithm in the CLSC scheme. For adversary $\mathcal{A}_1$, the challenger Ϲ sends system params to $\mathcal{A}_1$ but keeps s in secret. For adversary $\mathcal{A}_2$, the challenger Ϲ sends system params and s to $\mathcal{A}_2$.

(C2) The adversary asks the challenger Ϲ for a polynomial number of the queries.

(C3) The adversary chooses accepted sender $ID_S{}^*$, accepted receiver $ID_R{}^*$ (defined in Definition 2 and 3 below) and two random messages m_0 , m_1 to ask a challenging. The challenger Ϲ picks randomly $b \in \{0,1\}$ and computes σ^*. Then Ϲ returns σ^* to $\mathcal{A}$.

(C4) The adversary asks queries as done in step (C2), keeping $ID_S{}^*$ and $ID_R{}^*$ being accepted.

(C5) When terminating the game, $\mathcal{A}(\mathcal{A}_1$ or $\mathcal{A}_2)$ makes a guess bit b'. If $b' = b$, $\mathcal{A}$ wins the game.

The advantage of $\mathcal{A}$ for winning the game is defined as $IND\_Adv_A(k) = |pr[b = b'] - \frac{1}{2}|$.

Definition 2 (acceptable sender and receiver against $\mathcal{A}_1$ for confidentiality). For $\mathcal{A}_1$, Rx_S, Rpk_S, Rx_R and Rpk_R are always accepted. Then, the sender and receiver are accepted if none of the following condition holds.

(1) $\mathcal{A}_1$ either raises the query Rsk_R or Rd_R.

(2) $\mathcal{A}_1$ either asks the query Rsk_S or Rd_S.

(3) $\mathcal{A}_1$ raises query R*usc*$(\sigma^*, ID_S{}^*, ID_R{}^*)$.

Definition 3 (acceptable sender and receiver against $\mathcal{A}_2$ for confidentiality). For $\mathcal{A}_2$, Rd_S, and Rd_R are always accepted. The sender and receiver are accepted if none of the following condition holds.

(1) $\mathcal{A}_2$ either raises the query Rsk_R or Rx_R(Rpk_R).

(2) $\mathcal{A}_2$ either asks the query Rsk_S or Rx_S(Rpk_S).

(3) $\mathcal{A}_2$ raises query R*usc*$(\sigma^*, ID_S{}^*, ID_R{}^*)$.

Definition 4 (Unforgeability): A CLSC scheme is secure against unforgeability under adaptive chosen-messages attacks (EUF-CMA) only if the probability for attackers to win the following game is negligible with any polynomial time algorithm.

Steps of the **Game EUF-CMA** are described as follows.

(U1), (U2) The same as the steps (C1) and (C2) in Game IND-CCA2.

(U3) The adversary chooses accepted sender $ID_S{}^*$ (defined in Definition 5 and 6 below) and a user $ID_j{}^*$, outputs σ^* on a chosen messages m^*.

(U4) Ϲ executes unsigncryption algorithm with input as $(\sigma^*, ID_S{}^*, ID_R{}^*)$. If Ϲ outputs $m = m^*$, $\mathcal{A}$ wins the game.

The advantage of $\mathcal{A}$ for winning the game is defined as $EUF\_Adv_A(k) = |pr[m = m^*] - \frac{1}{2}|$.

Definition 5 (acceptable sender against $\mathcal{A}_1$ for unforgeability). For $\mathcal{A}_1$, Rx_S and Rpk_S are always accepted. The sender is accepted if none of the following condition holds.

(1) $\mathcal{A}_1$ either raises the query Rsk_S or Rd_S.

(2) $(\sigma^*, ID_S{}^*, ID_R{}^*)$ is not produced by signcryption algorithm with $(m^*, ID_S{}^*, ID_R{}^*)$.

Definition 6 (acceptable sender against $\mathcal{A}_2$ for unforgeability). For $\mathcal{A}_2$, Rd_S and Rd_R are always accepted. The sender is accepted if none of the following condition holds.

(1) $\mathcal{A}_2$ either raises the query Rsk_S or Rx_S(Rpk_S).

(2) $(\sigma^*, ID_S{}^*, ID_R{}^*)$ is not produced by signcryption algorithm with $(m^*, ID_S{}^*, ID_R{}^*)$.

Definition 7 (public key replacement (PKR) attack). A CLSC scheme resists public key replacement attack only if $\mathcal{A}_1$ cannot win Game IND-CCA2 and Game EUF-CMA.

Definition 8 (malicious but passive KGC (MPK) attack). A CLSC scheme resists malicious but passive KGC attack only if $\mathcal{A}_2$ cannot win Game IND-CCA2 and Game EUF-CMA.

Definition 9 (ephemeral secret leakage (ESL) attack). A CLSC scheme resists ESL attack means that even if the attacker $\mathcal{A}_1$ or $\mathcal{A}_2$ is allowed to ask Rr_i query, he cannot win Game IND-CCA2 and Game EUF-CMA.

2.3.3 Security definition

Definition 10 (secure CLSC scheme). A CLSC scheme is secure when it matches the following conditions.

(1) The sender generates the ciphertext σ with private keys of his own and public keys of the receiver, and the receiver recovers the correct plaintext m from σ with private keys of his own and public keys of the sender. Such correctness of a CLSC scheme can be defined as the following.

$$m = \mathrm{USC}(\mathrm{SC}\big(m, ID_i, sk_i, pk_i, ID_j, pk_j, system\ params\big), ID_i, pk_i, ID_j, sk_j, pk_j)$$

(2) $IND\_Adv_A(k)$ is negligible.

(3) $EUF\_Adv_A(k)$ is negligible.

3. Analysis on related CLSC schemes

In this section, we demonstrate the security weaknesses of several existing CLSC schemes. We find that all of them are vulnerable to ESL attack, MPK attack, PKR attack and fail to provide confidentiality and unforgeability under our security model.

3.1. Analysis on scheme [12] and [11]

Scheme [12] is briefly described as follows.

Setup: KGC chooses $s \in Z_q^*$ and computes $P_{pub} = sP$.

SetSecretValue: The user randomly chooses $x_i \in Z_q^*$, makes $PK_i = x_i P$ as public key.

ExtractPartialPrivateKey KGC computes partial private key as $d_i = sQ_i = sH_1(ID_i)$.

SetPrivateKey The user owns (x_i, d_i) as private key.

SetPublicKey The user owns PK_i as public key.

Signcrypt

In their scheme, when $ID_S \notin \emptyset$, $ID_R \notin \emptyset$, then $f(ID_S) = f(ID_R)$=1, algorithm runs in signcryption mode. The signer computes the ciphertext $c = (U, V, W)$ in the signcryption phase as follows:

The signer computes $r \in Z_q^*, U = rP, w = e\big(P_{pub}, Q_R\big)^r, h = H_2(U, w, rPK_R, ID_S, PK_S, ID_R, PK_R), V = m \oplus h, H = H_3(U, V, ID_S, PK_S, ID_R, PK_R)$, $H^{'} = H_4(U, V, ID_S, PK_S, ID_R, PK_R)$, $W = d_S + rH + x_S H^{'}$.

Attacks

The attacker, who gets the ephemeral private key r with query Rr_S, can compute h and get the message with $m = V \oplus h$. $\mathcal{A}_1$ can compute the partial private key as $d_S = W - rH - x_S H^{'}$ with queries Rr_S and Rx_S, $\mathcal{A}_2$ can compute the secret key as $x_S = (W - d_S - rH)/H^{'}$ with queries Rr_S and Rd_S.

Based on the proof above, scheme [12] cannot withstand ESL attack, MPK attack, PKR attack and fails to provide confidentiality and unforgeability.

Most phases in scheme [11] and [12] are the same. Similarly, Scheme [11] is insecure when the attacker knows random numbers r and τ.

3.2. Analysis on scheme [13]

Scheme [13] is briefly described as follows.

Setup: KGC chooses $s \in Z_q^*$ and computes $P_{pub} = sP$.

SetSecretValue: The user randomly chooses $x_{i \in Z_q^*}$, makes $PK_i = x_i P$ as public key.

ExtractPartialPrivateKey: KGC computes $d_i = sQ_i$, $Q_i = H_1(ID_i || PK_i)$.

SetPrivateKey The user owns (x_i, d_i) as private key.

Signcrypt

In [13], the signer computes the ciphertext $\sigma = (\tau, h, W, T)$ in the signcryption phase as follows:

Choose $r_1, r_2 \in Z_q^*$, compute $R_1 = r_1 P, R_2 = r_2 P, Q_R = H_1(ID_R || PK_R)$.

Compute $U = r_1 PK_R, V = e\big(r_2 Q_R, P_{pub}\big)P, K = H_2(ID_S, ID_R, R_1, R_2, U, V), \tau = Enk_K(m)$.

Compute $h = H_3(\tau, ID_S, ID_R, PK_S, PK_R, R_1, R_2, U, V), W = h\big(d_S + r_2 Q_S\big), T = hx_S + r_1$.

Attack

The attacker, who gets the ephemeral private keys r_1, r_2 with query Rr_S, can compute the symmetric key K and get the message $m = Dek_K(\tau)$. $\mathcal{A}_1$ can compute the partial private key as $d_S = h^{-1}W - r_2 Q_S$ with queries Rr_S. $\mathcal{A}_2$ can compute the secret key as $x_S = h^{-1}(T - r_1)$ with queries Rr_S.

Scheme [13], which cannot withstand ESL attack, MPK attack and PKR attack, fails to provide confidentiality and unforgeability.

3.3. Analysis on scheme [16]

Scheme [16] is briefly described as follows.

Setup: KGC computes $g_{pub} = g^s$, in which s is the master private key of KGC, g_{pub} is the public key of KGC.

SetSecretValue: The user randomly chooses y_i, makes $Y_i = g^{y_i}$ as public key.

ExtractPartialPrivateKey KGC randomly chooses x_{i0}, x_{i1} , computes $X_{i0} = g^{x_{i0}}, X_{i1} = g^{x_{i1}}, q_{i0} = H_1(ID_i, X_{i0}), q_{i1} = H_1(ID_i, X_{i0}, X_{i1}), d_{i0} = x_{i0} + sq_{i0}, d_{i1} = x_{i1} + sq_{i1}$. d_{i0} is set as partial private key.

SetPrivateKey The user owns (y_i, d_{i0}) as private key.

SetPublicKey The user owns $PK_i = (d_{i1}, X_{i0}, X_{i1}, Y_i)$ as public key.

The signer computes the ciphertext $C = (c_1, c_2, c_3)$ in the signcryption phase as follows:

Signcrypt

The signer Chooses the ephemeral private keys $r_1, r_2 \in Z_q^*$, computes $c_1 = g^{r_1}, c_2 = g^{r_2}, k_1 = (Y_R)^{r_1}, k_2 = ((X_{R0}){g_{pub}}^{q_{R0}})^{r_1}, d = H_3(m, c_2, ID_S, ID_R, PK_S), e = H_5(m, c_2, ID_S, ID_R, PK_S), v = d \cdot d_{S0} + e \cdot y_S + r_2, c_3 = H_4(k_1, k_2, ID_S, ID_R) \oplus (m || r_1 || v)$.

Attack

This scheme was insecure due to the computation of c_3. Suppose that the attacker gets the ephemeral private keys r_1, r_2 with query Rr_S, he can compute c_1, c_2, k_1, k_2 and get $m || r_1 || v$=$c_3 \oplus H_4(k_1, k_2, ID_S, ID_R)$. Since the attacker knows r_1, he can easily extract the message m and v in $m || r_1 || v$. Then, $\mathcal{A}_1$ can compute the partial private key as $d_{S0} =$

$\frac{v-r_2-e\cdot y_S}{d}$ with queries Rr_S and Ry_S. $\mathcal{A}_2$ can compute the secret key as $y_S=\frac{v-r_2-d\cdot d_{S0}}{e}$ with queries Rr_S and Rd_{S0}.

Scheme [16], which cannot withstand ESL attack, MPK attack and PKR attack, fails to provide confidentiality and unforgeability.

3.4. Analysis on scheme [22]

Scheme [22] is briefly described as follows.

Setup: KGC computes $y=g^x \, mod\, p$, in which x is the master private key of KGC, y is the public key of KGC.

SetSecretValue: The user randomly chooses x_i, makes $P_i=g^{x_i}$ as public key.

ExtractPartialPrivateKey KGC randomly chooses r_i, computes $R_i=g^{r_i}, s_i=r_i+xH_1(ID_i,R_i)$.

SetPrivateKey The user owns (x_i,d_i) as private key.

SetPublicKey The user owns (P_i,R_i) as public key.

The signer computes the ciphertext $C=(c_1,c_2,c_3)$ in the signcryption phase as follows:

Signcrypt

The signer Chooses r_S, compute $c_1=g^{r_S}\, mod\, p$, $k_S=H_2(ID_S,P_S,R_S,y)$, $k_R=H_2(ID_R,P_R,R_R,y)$, $h_R=H_1(ID_R,R_R)$, $\xi=\left(P_R{}^{k_R}R_R y^{h_R}\right)^{r_S} mod\, p$, $c_2=H_3(\xi)\oplus m$, $h=H_4(ID_S,P_S,R_S,c_1,\mathrm{c}_2,\xi,m)$, $c_3=[(k_S x_S+d_S)/(r_S+h)]\, mod\, q$.

Attack

The attacker, who gets the ephemeral private keys r_S with query Rr_S, can compute the message m with $\xi=\left(P_R{}^{k_R}R_R y^{h_R}\right)^{r_S} mod\, p$ and $m=H_3(\xi)\oplus c_2$. $\mathcal{A}_1$ can compute the partial private key as $d_S=c_3(r_S+h)-k_S x_S$ with queries Rr_S and Rx_S. $\mathcal{A}_2$ can compute the secret key as $x_S=(c_3(r_S+h)-d_S)/k_S$ with queries Rr_S and Rd_S.

Scheme [22], which cannot withstand ESL attack, MPK attack and PKR attack, fails to provide confidentiality and unforgeability.

4. Our proposed CLSC scheme

Motivated by the structure of previous CLSC schemes, we propose a novel CLSC scheme without pairing based on ECC as shown in **Fig. 3**. Our scheme consists of two phases: registration, signcrypt &unsigncrypt.

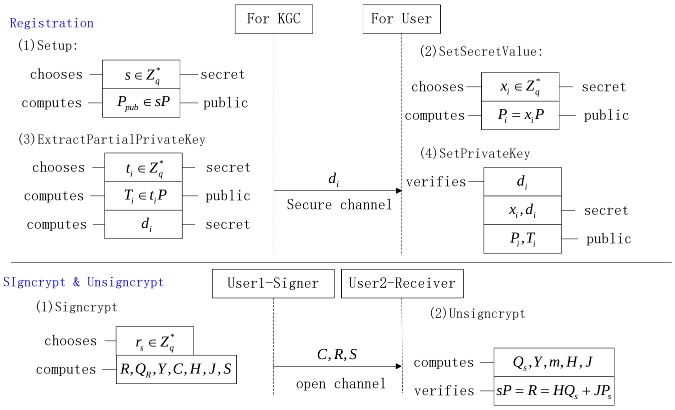

Fig. 3. Our proposed pairing-free CLSC schemes

(1)Registration phase

Setup: KGC chooses $s \in Z_q^*$ and computes $P_{pub} = sP$.

SetSecretValue: The user randomly chooses $x_i \in Z_q^*$, makes $P_i = x_iP$ as public key.

ExtractPartialPrivateKey KGC randomly chooses $t_i \in Z_q^*$, computes $T_i = t_iP, l_i = H_0(ID_i, T_i, P_i) \in (0,1)^k, d_i = (t_i + s\ l_i) mod\ q$. d_i is set as partial private key.

SetPrivateKey The user owns (x_i, d_i) as private key.

SetPublicKey The user owns $PK_i = (P_i, T_i)$ as public key.

The user verifies whether $d_iP = T_i + H_0(ID_i, T_i, P_i)P_{pub} = Q_i$ or not.

(2) Signcrypt &Unsigncrypt phase

Signcrypt

The signer chooses randomly $r_S \in Z_q^*$, computes

$R = r_S(H_4(ID_S, P_S)Q_S + H_4(ID_R, P_R)P_S), Q_R = T_R + H_0(ID_R, T_R, P_R)P_{pub},$

$Y = r_S(d_S(H_4(ID_S, P_S) + x_S H_4(ID_R, P_R)\ )(H_4(ID_S, P_S)Q_R + H_4(ID_R, P_R)P_R),$

$c = m \oplus H_1(Y), H = H_2(m, c, R, Y, Q_S, Q_R),\ \ J = H_3(m, c, R, Y, P_S, P_R),$

$S = r_S\big(d_S H_4(ID_S, P_S) + x_S H_4(ID_R, P_R)\big) + d_S H + x_S J.$

Then, the signer transmits ciphertext $\sigma = (c, R, S)$ to the receiver.

Unsigncrypt

After receiving $\sigma = (c, R, S)$, the receiver executes the unsigncryption algorithm as follows.

The receiver computes $Q_S = T_S + H_0(ID_S, T_S, P_S)P_{pub}$ $, Y = R(d_R(H_4(ID_S, P_S) + x_R H_4(ID_R, P_R)\ ), m = c \oplus H_1(Y),\ \ H = H_2(m, c, R, Y, Q_S, Q_R),\ \ J = H_3(m, c, R, Y, P_S, P_R),$

The receiver will accept m if $SP = R + HQ_S + JP_S$ holds.

5. Analysis of our proposed CLSC scheme

5.1. Security analysis

According to the definition in section 2.3.3, our CLSC scheme is secure under the GDH assumption and DL problem.

Theorem 1 Our scheme is correct.

Proof. Our scheme is correct because of the following.

After receiving $\sigma = (c, R, S)$, the receiver computes $Q_S = T_S + H_0(ID_S, T_S, P_S)P_{pub} = d_SP,$

$Y = R(d_R(H_4(ID_S, P_S) + x_R H_4(ID_R, P_R)\ )$

$= r_S(H_4(ID_S, P_S)Q_S + H_4(ID_R, P_R)P_S)(d_R(H_4(ID_S, P_S) + x_R H_4(ID_R, P_R)\ )$

$= r_S(H_4(ID_S, P_S)d_S + H_4(ID_R, P_R)x_S)(d_R(H_4(ID_S, P_S) + x_R H_4(ID_R, P_R)\ )P$

$= r_S(H_4(ID_S, P_S)d_S + H_4(ID_R, P_R)x_S)((H_4(ID_S, P_S)Q_R + H_4(ID_R, P_R)P_R\ )$

So, the receiver recovers m with $m = c \oplus H_1(Y)$.

Then, the receiver verifies m with

$H = H_2(m, c, R, Y, Q_S, Q_R),\ \ J = H_3(m, c, R, Y, P_S, P_R),$

$SP = (r_S\big(d_S H_4(ID_S, P_S) + x_S H_4(ID_R, P_R)\big) + d_S H + x_S J)P$

$= (r_S\big(d_S P H_4(ID_S, P_S) + x_S P H_4(ID_R, P_R)\big) + d_S PH + x_S PJ)$

$= r_S(H_4(ID_S, P_S)Q_S + H_4(ID_R, P_R)P_S) + HQ_S + JP_S$

$= R + HQ_S + JP_S.$

Theorem 2 Our scheme provides confidentiality under the GDH assumption.

This theorem can be derived from the **Lemma1** and **Lemma 2**.

Suppose $H_0(*)$, $H_1(*)$, $H_2(*)$, $H_3(*)$, $H_4(*)$ are random oracles owned by C. Assume that $\mathcal{A}_1$ makes at most q_i queries to H_i $(0 \le i \le 4)$ respectively, q_c queries to Create(ID_i), q_d queries to Rd_i, q_x queries to Rx_i, q_{pk} queries to Rpk_i, q_{sk} queries to Rsk_i, q_r queries to Rr_i, q_{sc} queries to R$_{sc}$ and q_{usc} queries to R$_{usc}$. Assume also that bounded running time of query H_i $(0 \le i \le 4)$ is t_i , Create(ID_i) is t_c , Rd_i is t_d , Rx_i is t_x , Rpk_i is t_{pk} , Rsk_i is t_{sk} , Rr_i is t_r , R$_{sc}$ is t_{sc} and R$_{usc}$ is t_{usc} .

The challenger C maintains the query lists for consistency.

L_0 : a tuple of (ID_i, T_i, P_i, h_0^i).

L_1 : a tuple of (Y, h_1^i).

L_2 : a tuple of $(m, c, R, Y, Q_i, Q_j, h_2^i)$.

L_3 : a tuple of $(m, c, R, Y, P_i, P_j, h_3^i)$.

L_4 : a tuple of (ID_i, P_i, h_4^i).

L_C :a tuple of $(ID_i, d_i, x_i, P_i, T_i, r_i, h_0^i)$

L_{sc} :a tuple of (m, ID_i, ID_j, σ), $\sigma = (c, R, S)$

Lemma 1.

Given an instance of the GDH problem: For unknown $A, B \in Z_q^*$, by giving $P, AP, BP, P \in E/\ Fp$ and an oracle DDH, compute ABP.

Suppose $\mathcal{A}_1$ win the **Game IND-CCA2** with advantage ε and running time t, then an algorithm Γ can be constructed to solve the above instance of the GDH problem with advantage ε' and running time τ by interacting with $\mathcal{A}_1$.

$$\varepsilon' = \frac{2}{q_c(q_c-1)} \cdot (\frac{q_c-2}{q_c})^{q_d} \cdot (\frac{q_c-2}{q_c})^{q_{sk}} \cdot [\frac{1}{2} + \frac{1}{2}\left(\frac{2^k-1}{2^k}\right)^{q_{usc}}] \cdot \varepsilon$$

$$\tau \le \sum_{i=0}^{4} q_i\, t_i + q_c\, t_c + q_d\, t_d + q_x\, t_x + q_{pk}\, t_{pk} + q_r\, t_r + q_{sc}\, t_{sc} + q_{usc}\, t_{usc} + t + t_{CP}$$

Proof. To interact with $\mathcal{A}_1$, algorithm Γ simulates as C and runs the following steps to solve the above instance of the GDH problem with the help of $\mathcal{A}_1$.

(C1) Γ executes the SETUP algorithm and sends system params to $\mathcal{A}_1$.

(C2) Suppose that Γ will choose accepted sender S with identity $ID_S{}^*$ and accepted receiver R with identity $ID_R{}^*$ for challenge in the next step. $\mathcal{A}_1$ asks the Γ for a polynomial number of the queries.

$\boldsymbol{H_0}$ **query:** On receiving (ID_i, T_i, P_i), Γ performs as follows:

1) If L_0 contains a tuple of (ID_i, T_i, P_i, h_0^i), Γ returns h_0^i to $\mathcal{A}_1$.

2) Otherwise,

a) If $i \ne S, R$, Γ randomly chooses h_0^i and inserts (ID_i, T_i, P_i, h_0^i) to L_0 and returns h_0^i to $\mathcal{A}_1$.

b) Otherwise, Γ gets h_0^i from L_C , inserts (ID_i, T_i, P_i, h_0^i) to L_0 and returns h_0^i to $\mathcal{A}_1$.

Create($\boldsymbol{ID_i}$): On receiving (ID_i), Γ performs as follows:

1) If L_C contains a tuple of $(ID_i, d_i, x_i, P_i, T_i, r_i, h_0^i)$.

a) If $i \ne S, R$, Γ returns all the elements of the tuple to $\mathcal{A}_1$.

b) Otherwise, Γ returns $(ID_i, \perp, x_i, P_i, T_i, r_i, h_0^i)$ to $\mathcal{A}_1$.

2) Otherwise,

a) If $i \ne S, R,$ then Γ randomly chooses x_i, t_i, r_i, computes $P_i = x_i P$, $T_i = t_i P$, asks H_0 query to get h_0^i, then computes $d_i = t_i + h_0^i s$, Γ inserts $(ID_i, d_i, x_i, P_i, T_i, r_i, h_0^i)$ to L_C and returns$(ID_i, d_i, x_i, P_i, T_i, r_i, h_0^i)$ to $\mathcal{A}_1$.

b) Otherwise, Γ randomly chooses r_i, x_i, h_0^i, computes $P_i = x_i P$, $T_S = AP - h_0^S P_{pub}$, $T_R = BP - h_0^R P_{pub}$, Γ inserts $\left(ID_i, \perp, x_i, P_i, T_i, r_i, h_0^i\right)$ to L_C, inserts (ID_i, T_i, P_i, h_0^i) to L_0 and returns $\left(ID_i, \perp, x_i, P_i, T_i, r_i, h_0^i\right)$ to $\mathcal{A}_1$.

All the following queries should be asked after Create(*ID_i*)

$\boldsymbol{H_1}$ query: On receiving (Y), Γ performs as follows:

If L_1 contains a tuple of $\left(Y, h_1^i\right)$, Γ returns h_1^i to $\mathcal{A}_1$. Otherwise, Γ randomly chooses h_1^i and inserts $\left(Y, h_1^i\right)$ to L_1 and returns h_1^i to $\mathcal{A}_1$.

$\boldsymbol{H_2}$ query: On receiving $\left(m, c, R, Y, Q_i, Q_j\right)$, Γ performs as follows:

If L_2 contains a tuple of $\left(m, c, R, Y, Q_i, Q_j, h_2^i\right)$, Γ returns h_2^i to $\mathcal{A}_1$. Otherwise, Γ randomly chooses h_2^i and inserts $\left(m, c, R, Y, Q_i, Q_j, h_2^i\right)$ to L_2 and returns h_2^i to $\mathcal{A}_1$.

$\boldsymbol{H_3}$ query: On receiving $\left(m, c, R, Y, P_i, P_j\right)$, Γ performs as follows:

If L_3 contains a tuple of $\left(m, c, R, Y, P_i, P_j, h_3^i\right)$, Γ returns h_3^i to $\mathcal{A}_1$. Otherwise, Γ randomly chooses h_3^i and inserts $\left(m, c, R, Y, P_i, P_j, h_3^i\right)$ to L_3 and returns h_3^i to $\mathcal{A}_1$.

$\boldsymbol{H_4}$ query: On receiving (ID_i, P_i), Γ performs as follows:

If L_4 contains a tuple of $\left(ID_i, P_i, h_4^i\right)$, Γ returns h_4^i to $\mathcal{A}_1$. Otherwise, Γ randomly chooses h_4^i and inserts $\left(ID_i, P_i, h_4^i\right)$ to L_4 and returns h_4^i to $\mathcal{A}_1$.

R*d_i* query: On receiving ID_i, Γ performs as follows:

1) If $i \neq S, R$, Γ returns d_i from L_C to $\mathcal{A}_1$.

2) Otherwise, the game is aborted.

R*x_i* query:. On receiving ID_i, Γ returns x_i from L_C to $\mathcal{A}_1$.

R*sk_i* query: should be asked after Create(*ID_i*). On receiving ID_i, Γ performs as follows:

1) If $i \neq S, R$, Γ returns (d_i, x_i) from L_C to $\mathcal{A}_1$.

2) Otherwise, the game is aborted.

R*pk_i* query: On receiving ID_i, $\mathcal{A}_1$ randomly chooses $x_i^{'}$, computes $P_i^{'} = x_i^{'} P$, Γ updates all the tuples with $x_i = x_i^{'}$, $P_i = P_i^{'}$.

R*r_i* query: Γ returns r_i from L_C to $\mathcal{A}_1$.

R_{*sc*}(m, ID_i, ID_j) **query:** Γ performs as follows:

1) If $i \neq S, R$,

According to the queries Create(*ID_i*) and Create(*ID_j*), $\mathcal{A}_1$can get $ID_i, d_i, x_i, r_i, ID_j, P_j, T_j$, then Γ executes the signcryption algorithm and returns $(\sigma = (c, R, S), ID_i, ID_j)$ to $\mathcal{A}_1$.

2) If $i = S\ or\ i = R, j \neq S, R$,

a) $\mathcal{A}_1$ gets h_0^i, h_0^j from L_0, gets ID_j , d_j, x_j, T_j, P_j from Create(*ID_j*), gets ID_i, r_i, T_i, P_i from Create(*ID_i*), where $T_i = I \cdot P - h_0^i P_{pub}$ (*I*=*A* when $i = S$, *I*=*B* when $i = R$), gets h_4^i, h_4^j from L_4 .

b) Then Γ computes $Q_j = T_j + h_0^j P_{pub}$, $Q_i = T_i + h_0^i P_{pub} = I \cdot P$, $R = r_i\left(h_4^i Q_i + h_4^j P_i\right)$, $Y = R\left(d_j h_4^i + x_j h_4^j\right)$.

c) $\mathcal{A}_1$ gets h_1^i from L_1 .

d) Γ computes $c = m \oplus h_1^i$.

e) $\mathcal{A}_1$ gets h_2^i with (m, c, R, Y, Q_i, Q_j) in H_2 query.

f) Γ randomly chooses S, computes $h_3^i = \frac{SP - R - h_2^i I \cdot P}{P_i}$. If the chosen S and h_3^i already exist in L_{sc} with $(m, ID_i, ID_j, (c, R, S))$ and L_3 , then Γ chooses another S , computes h_3^i.

g) Γ inserts $\left(m, c, R, Y, P_i, P_j, h_3^i\right)$ to L_3 and $\left(m, ID_i, ID_j, (c, R, S)\right)$ to L_{sc} , returns $\sigma = (c, R, S)$ to $\mathcal{A}_1$.

3) If $i = S\ , j = R$,

a) $\mathcal{A}_1$ gets h_0^i, h_0^j from L_0 , gets $ID_i, r_i, T_i, P_i, ID_j\ , T_j, P_j$ from Create(*ID*) and Create(*ID*), where $T_i = A \cdot P - h_0^i P_{pub}, T_j = B \cdot P - h_0^j P_{pub}$, gets h_4^i, h_4^j from L_4 .

b) Then Γ computes $Q_i = T_i + h_0^i P_{pub} = A \cdot P, Q_j = T_j + h_0^j P_{pub} = B \cdot P$, $R = r_i\left(h_4^i Q_i + h_4^j P_i\right)$, randomly chooses Y. Since Y is chosen randomly, $\mathcal{A}_1$ cannot verify the validity of c.

The remaining steps only differ in the step f) with that of the situation above when computing h_3^i with $h_3^i = \frac{SP-R-h_2^i A \cdot P}{P_i}$.

R*usc*(σ, ID_i, ID_j) query:

1) If $\mathrm{i} \neq \mathrm{S}, \mathrm{R}$,

According to the queries Create(ID_i) and Create(ID_j), $\mathcal{A}_1$can get $ID_i, d_i, x_i, r_i, ID_j, P_j, T_j$, then Γ gets h_4^i, h_4^j from L_4 and computes $Y = r_i\left(d_i h_4^i + x_i h_4^j\ \right)\left(h_4^i Q_R + h_4^j P_R\right)$, $m = c \oplus H_1(Y)$, returns m to $\mathcal{A}_1$.

2) If $i = S$ or $i = R$ and $j \neq S, R$

According to the queries Create(ID_i) and Create(ID_j), $\mathcal{A}_1$can get $ID_i, P_i, T_i, ID_j, d_j, x_j$, then Γ executes the unsigncryption algorithm and returns m to $\mathcal{A}_1$

3) If $\mathrm{i} = \mathrm{S}, j = \mathrm{R}$,

a) if $\left(\sigma, ID_i, ID_j\right)$ exists in L_{sc} , Γ returns m in the list to $\mathcal{A}_1$.

b) Otherwise, Γ rejects R$_{usc}(\sigma, ID_i, ID_j)$ query.

RDDH($\boldsymbol{aP}, \boldsymbol{bP}, \boldsymbol{cP}$) query:

The oracle DDH outputs 1 if $abP = cP$, otherwise 0.

(C3) The adversary chooses accepted sender $ID_S{}^*$, accepted receiver $ID_R{}^*$ and m_0 , m_1 to ask a challenging. The challenger C performs as follows

1) gets $h_0^S, h_0^R, h_4^S, h_4^R, ID_S, r_S, x_S, T_S, P_S, ID_R\ , x_R, T_R, P_R$ from L_0 , L_4 , Create(ID_S) and Create(ID_R), where $Q_S = A \cdot P, Q_R = B \cdot P$, computes $R = r_S\left(h_4^S Q_S + h_4^R P_S\right)$

2) computes Y^* with the candidate solution of ABP, gets h_1^S from L_1 .

3) picks randomly $b \in \{0,1\}$, computes $c^* = m_b \oplus h_1^S$.

4) gets h_2^S from L_2 .

5) randomly chooses S, computes $h_3^S = \frac{SP-R-h_2^S A \cdot P}{P_S}$.

6) inserts $\left(m_b, c^*, R, Y^*, P_S, P_R, h_3^S\right)$ to L_3 , returns ($\sigma^* = (c^*, R, S), ID_S{}^*, ID_R{}^*$) to $\mathcal{A}_1$.

(C4) The adversary asks queries as done in step (C2), keeping $ID_S{}^*$ and $ID_R{}^*$ being accepted.

(C5) As $\mathcal{A}_1$ win the **Game IND-CCA2** by guessing $b' = b$ with advantage ε, with the help of $\mathcal{A}_1$, Γ can compute $h_1^S = m_b \oplus c^*$, get Y^* in L_1 , ask RDDH query with RDDH($R, h_4^S Q_R + h_4^R P_R, Y^*$)=1, then Γ gets $C \cdot P$ with running time of $t_{CP} \approx 4T_{mul} + 3T_{add}$, in which T_{mul} is the time for one scalar multiplication operation over elliptic curve and T_{add} the point addition operation over elliptic curve.

$$C \cdot P = d_s d_R \cdot P = \frac{\frac{Y^*}{r_s} - h_4^S h_4^R x_R\ A \cdot P - h_4^S h_4^R x_S B \cdot P - h_4^R h_4^R x_R P_S}{h_4^S h_4^S}$$

With the above description, Γ wins to solve the GDH problem only if when choosing ${ID_S}^*$ and ${ID_R}^*$ for challenge (i.e. event ${E_1}'$ occurs), the game is completed. But, Γ will terminate the game when any of the events E_1, E_2, E_3, E_4 occurs.

E_1: $\mathcal{A}_1$ does not choose both S with identity of ${ID_S}^*$and R with identity ${ID_R}^*$ for challenge.

E_2: $\mathcal{A}_1$ asks Rd_i query with ${ID_S}^*$ or ${ID_R}^*$,

$$\Pr\,[E_2] = \frac{C_2^1}{C_{q_c}^1} + \frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \frac{C_2^1}{C_{q_c}^1} + \cdots + \underbrace{\frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \ldots \cdot \frac{C_{q_c-2}^1}{C_{q_c}^1}}_{q_d-1} \cdot \frac{C_2^1}{C_{q_c}^1} = 1 - (\frac{q_c-2}{q_c})^{q_d}.$$

E_3: $\mathcal{A}_1$ asks Rsk_i query with ${ID_S}^*$ or ${ID_R}^*$,

$$\Pr\,[E_3] = \frac{C_2^1}{C_{q_c}^1} + \frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \frac{C_2^1}{C_{q_c}^1} + \cdots + \underbrace{\frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \ldots \cdot \frac{C_{q_c-2}^1}{C_{q_c}^1}}_{q_{sk}-1} \cdot \frac{C_2^1}{C_{q_c}^1} = 1 - (\frac{q_c-2}{q_c})^{q_{sk}}.$$

E_4: $\mathcal{A}_1$ asks R$_{usc}$ query with $(m_b, {ID_S}^*, {ID_R}^*)$,

$$\Pr\,[E_4] = \frac{1}{2} \cdot \left[\frac{1}{2^k} + \left(1 - \frac{1}{2^k}\right) \cdot \frac{1}{2^k} + \cdots + \underbrace{\left(1 - \frac{1}{2^k}\right) \cdot \left(1 - \frac{1}{2^k}\right) \cdot \ldots \cdot \left(1 - \frac{1}{2^k}\right)}_{q_{usc}-1} \cdot \frac{1}{2^k}\right] = \frac{1}{2} - \frac{1}{2}\left(\frac{2^k-1}{2^k}\right)^{q_{usc}}$$

${E_1}'$: $\mathcal{A}_1$ choose both ${ID_S}^*$and ${ID_R}^*$ for challenge.Pr $[{E_1}'] = \frac{1}{C_{q_c}^2} = \frac{2}{q_c(q_c-1)}$.

Then, if $\mathcal{A}_1$ win the **Game IND-CCA2** with advantage ε and running time t, then an algorithm Γ can be constructed to solve the GDH problem with advantage ε' by interacting with $\mathcal{A}_1$.

$$\begin{aligned} \varepsilon' &= \Pr[{E_1}'] \cdot (1 - \Pr\,[E_2]) \cdot (1 - \Pr\,[E_3]) \cdot (1 - \Pr\,[E_4]) \cdot \varepsilon \\ &= \frac{2}{q_c(q_c-1)} \cdot \left(\frac{q_c-2}{q_c}\right)^{q_d} \cdot \left(\frac{q_c-2}{q_c}\right)^{q_{sk}} \cdot \left[\frac{1}{2} + \frac{1}{2}\left(\frac{2^k-1}{2^k}\right)^{q_{usc}}\right] \cdot \varepsilon \end{aligned}$$

$$\tau \le \sum_{i=0}^{4} q_i\, t_i + q_c\, t_c + q_d\, t_d + q_x\, t_x + q_{pk}\, t_{pk} + q_r\, t_r + q_{sc}\, t_{sc} + q_{usc}\, t_{usc} + t + t_{CP}$$

Lemma 2.

Suppose $\mathcal{A}_2$ win the **Game IND-CCA2** with advantage ε and running time t, then an algorithm Γ can be constructed to solve the instance of the GDH problem in **Lemma 1** with advantage $\varepsilon'$and running time τ by interacting with $\mathcal{A}_1$.

$$\varepsilon' = \frac{2}{q_c(q_c-1)} \cdot \left(\frac{q_c-2}{q_c}\right)^{q_{sk}} \cdot \left(\frac{q_c-2}{q_c}\right)^{q_x} \cdot \left(\frac{q_c-2}{q_c}\right)^{q_{pk}} \cdot \left[\frac{1}{2} + \frac{1}{2}\left(\frac{2^k-1}{2^k}\right)^{q_{usc}}\right] \cdot \varepsilon$$

$\tau \le \sum_{i=0}^{4} q_i\, t_i + q_c\, t_c + q_d\, t_d + q_x\, t_x + q_{pk}\, t_{pk} + q_r\, t_r + q_{sc}\, t_{sc} + q_{usc}\, t_{usc} + t + t_{CP}$

Proof. To interact with $\mathcal{A}_2$, algorithm Γ runs the following steps to solve the instance of the GDH problem .

(C1) Γ executes the SETUP algorithm and sends system params and master secret key s to $\mathcal{A}_2$.

(C2) $\mathcal{A}_2$ asks Γ for a polynomial number of the queries as shown in **Lemma 1**, Γ answers the following queries differently.

***H_0* query:** On receiving (ID_i, T_i, P_i), Γ performs as follows:

1) If L_0 contains a tuple of (ID_i, T_i, P_i, h_0^i), Γ returns h_0^i to $\mathcal{A}_1$.

2) Otherwise, Γ randomly chooses h_0^i and inserts (ID_i, T_i, P_i, h_0^i) to L_0 and returns h_0^i to $\mathcal{A}_1$.

Create(*ID_i*): On receiving (ID_i), Γ performs as follows:

1) If L_C contains a tuple of $(ID_i, d_i, x_i, P_i, T_i, r_i, h_0^i)$.

a) If $i \neq S, R$, Γ returns all the elements of the tuple to $\mathcal{A}_2$.

b) Otherwise, Γ returns $(ID_i, d_i, \perp, P_i, T_i, r_i, h_0^i)$ to $\mathcal{A}_2$.

2) Otherwise,

a) If $i \neq S, R$, Γ randomly chooses x_i, t_i, r_i, computes $P_i = x_i P$, $T_i = t_i P$, asks H_0 query to get h_0^i , then computes $d_i = t_i + h_0^i s$, Γ inserts $(ID_i, d_i, x_i, P_i, T_i, r_i, h_0^i)$ to L_C and returns$(ID_i, d_i, x_i, P_i, T_i, r_i, h_0^i)$ to $\mathcal{A}_2$.

b) Otherwise, Γ randomly chooses t_i , r_i , sets $P_S = AP, P_R = BP$, gets h_0^i from L_0 , computes $d_i = t_i + h_0^i s$, inserts $(ID_i, d_i, \perp, P_i, T_i, r_i, h_0^i)$ to L_C and returns $(ID_i, d_i, \perp, P_i, T_i, r_i, h_0^i)$ to $\mathcal{A}_2$.

All the following queries should be asked after Create(ID_i).

Rx_i query: On receiving ID_i, Γ performs as follows:

1) If $i \neq S, R$, Γ returns x_i from L_C to $\mathcal{A}_2$.

2) Otherwise, the game is aborted.

Rd_i query: On receiving ID_i, Γ returns d_i from L_C to $\mathcal{A}_2$.

Rpk_i query: On receiving ID_i, Γ performs as follows:

1) If $i \neq S, R$, $\mathcal{A}_2$ randomly chooses x_i', computes $P_i' = x_i' P$, Γ updates all the tuples with $x_i = x_i'$, $P_i = P_i'$

2) Otherwise, the game is aborted.

R$_{sc}$ (m, ID_i, ID_j) **query:** Γ performs the same steps as shown in **Lemma 1** except the following steps.

1) If $i \neq S, R$, Γ executes the same steps as shown in **Lemma 1**.

2) If $i = S\ or\ i = R, j \neq S, R,$

a) $\mathcal{A}_2$ gets h_0^i, h_0^j from L_0 , gets ID_j , d_j, x_j, T_j, P_j from Create(ID_j), gets ID_i, r_i, T_i, P_i from Create(ID_i), where $P_i = I \cdot P$ (I=A when $i = S$, I=B when $i = R$), gets h_4^i, h_4^j from L_4 .

b) Then Γ computes $Q_i = T_i + h_0^i P_{pub}, Q_j = T_j + h_0^j P_{pub},$, $R = r_i(h_4^i Q_i + h_4^j \cdot I \cdot P)$, $Y = R(d_j h_4^i + x_j h_4^j)$.

f) computes h_3^i with $h_3^i = \frac{SP - R - h_2^i Q_i}{I \cdot P}$

3) If $i = S$, $j = R$,

a) $\mathcal{A}_2$ gets h_0^i, h_0^j from L_0 , gets $ID_i, r_i, T_i, P_i, ID_j$, T_j, P_j from Create(ID_i) and Create(ID_j), where $P_i = A \cdot P$, $P_j = B \cdot P$, gets h_4^i, h_4^j from L_4 .

b) Then Γ computes $Q_i = T_i + h_0^i P_{pub}, Q_j = T_j + h_0^j P_{pub}$, $R = r_i(h_4^i Q_i + h_4^j \cdot A \cdot P)$, randomly chooses Y.

f) computes h_3^i with $h_3^i = \frac{SP - R - h_2^i Q_i}{A \cdot P}$.

(C3) Γ executes the same steps as (C3) shown in **Lemma 1** except the following steps.

1) $\mathcal{A}_2$ gets h_0^S, h_0^R from L_0 , gets $ID_S, r_S, d_S, T_S, P_S, ID_R$, T_R, P_R from Create(ID_S) and Create(ID_R), where $P_S = A \cdot P, P_R = B \cdot P$, gets h_4^S, h_4^R from L_4 , computes $Q_S = T_S + h_0^S P_{pub}, Q_R = T_R + h_0^R P_{pub}$, $R = r_S(h_4^S Q_S + h_4^R P_S)$.

5) computes $h_3^S = \frac{SP - R - h_2^S Q_S}{A \cdot P}$

(C4) The adversary asks queries as done in step (C2), keeping $ID_S{}^*$ and $ID_R{}^*$ being accepted.

(C5) As $\mathcal{A}_2$ win the **Game IND-CCA2** by guessing $b' = b$ with advantage ε, then Γ can compute h_1^S= $m_b \oplus c^*$, get Y^* in L_1 , ask RDDH query with RDDH($R, h_4^S Q_R + h_4^R P_R, Y^*$)=1, then Γ gets $C \cdot P$ with running time of $t_{CP} \approx 4T_{mul} + 3T_{mul}$.

$$C \cdot P = x_S x_R \cdot P = \frac{\frac{Y^*}{r_S} - h_4^S h_4^S d_R\, Q_S - h_4^S h_4^R d_R A \cdot P - d_S h_4^S h_4^R B \cdot P}{h_4^R h_4^R}$$

When event $E_1{}'$ occurs, Γ will terminate the game when any of the events E_1, E_3, E_4, E_5, E_6 occurs.

E_5: $\mathcal{A}_2$ asks Rx_i query with $ID_S{}^*$ or $ID_R{}^*$,

$$\Pr\,[E_5] = \frac{C_2^1}{C_{q_c}^1} + \frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \frac{C_2^1}{C_{q_c}^1} + \cdots + \underbrace{\frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \ldots \cdot \frac{C_{q_c-2}^1}{C_{q_c}^1}}_{q_x - 1} \cdot \frac{C_2^1}{C_{q_c}^1} = 1 - (\frac{q_c-2}{q_c})^{q_x}.$$

E_6: $\mathcal{A}_2$ asks Rpk_i query with $ID_S{}^*$ or $ID_R{}^*$,

$$\Pr\,[E_6] = \frac{C_2^1}{C_{q_c}^1} + \frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \frac{C_2^1}{C_{q_c}^1} + \cdots + \underbrace{\frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \frac{C_{q_c-2}^1}{C_{q_c}^1} \cdot \ldots \cdot \frac{C_{q_c-2}^1}{C_{q_c}^1}}_{q_{pk} - 1} \cdot \frac{C_2^1}{C_{q_c}^1} = 1 - (\frac{q_c-2}{q_c})^{q_{pk}}.$$

Then, if $\mathcal{A}_2$ win the **Game IND-CCA2** with advantage ε and running time t, then an algorithm Γ can be constructed to solve the GDH problem with advantage ε' and running time τ by interacting with $\mathcal{A}_2$.

$$\varepsilon' = \Pr[E_1{}']) \cdot (1 - \Pr\,[E_3]) \cdot (1 - \Pr\,[E_4]) \cdot (1 - \Pr\,[E_5]) \cdot (1 - \Pr\,[E_6]) \cdot \varepsilon$$
$$= \frac{2}{q_c(q_c-1)} \cdot (\frac{q_c-2}{q_c})^{q_{sk}} \cdot (\frac{q_c-2}{q_c})^{q_x} \cdot (\frac{q_c-2}{q_c})^{q_{pk}} \cdot [\frac{1}{2} + \frac{1}{2}\left(\frac{2^k-1}{2^k}\right)^{q_{usc}}] \cdot \varepsilon$$

$$\tau \le \textstyle\sum_{i=0}^{4} q_i\, t_i + q_c\, t_c + q_d\, t_d + q_x\, t_x + q_{pk}\, t_{pk} + q_r\, t_r + q_{sc}\, t_{sc} + q_{usc}\, t_{usc} + t + t_{CP}$$

According to **Lemma1** and **Lemma 2**, if $\mathcal{A}$ win the **Game IND-CCA2** in polynomial time, Γ can solve the GDH problem, which is contradictory with the security assumption of GDH problem. Then, we conclude that $\mathcal{A}$ cannot win the **Game IND-CCA2** and $IND\_Adv_A(k)$ is negligible. Therefore, our scheme can provide confidentiality under the GDH assumption.

Theorem 3 Our scheme provides unforgeability under the DL problem.

This theorem can be derived from the **Lemma3** and **Lemma 4**.

Lemma 3

Given an instance of the DL problem: For unknown $A \in Z_q^*$, by giving $AP, P \in E/\,Fp$, compute A.

Suppose $\mathcal{A}_1$ win the **Game EUF-CMA** with advantage ε and running time t, then an algorithm Γ can be constructed to solve the above instance of the DL problem with advantage $\varepsilon'$and running time τ by interacting with $\mathcal{A}_1$.

$$\varepsilon' = \frac{1}{q_c} \cdot (\frac{q_c-1}{q_c})^{q_d} \cdot (\frac{q_c-1}{q_c})^{q_{sk}} \cdot \left(\frac{2^k-1}{2^k}\right)^{q_{sc}} \cdot \frac{1}{\sqrt{q_2}} \cdot \varepsilon$$

$$\tau \le \textstyle\sum_{i=0}^{4} q_i\, t_i + q_c\, t_c + q_d\, t_d + q_x\, t_x + q_{pk}\, t_{pk} + q_r\, t_r + q_{sc}\, t_{sc} + q_{usc}\, t_{usc} + 2t + t_{ds}$$

Proof. To interact with $\mathcal{A}_1$, algorithm Γ simulates as C and runs the following steps to solve the above instance of the DL problem with the help of $\mathcal{A}_1$.

(U1) Γ executes the SETUP algorithm and sends system params to $\mathcal{A}_1$.

(U2) Suppose that Γ will choose accepted sender S with identity $ID_S{}^*$ and a user $ID_j{}^*$for challenge. $\mathcal{A}_1$ asks the Γ for a polynomial number of the queries as shown in Lemma 1. Queries only contains conditions of $i \ne S$ and $i = S$, where receiver R should not be considered and specified.

(U3) The adversary $\mathcal{A}_1$ chooses accepted sender $ID_S{}^*$ and a user $ID_j{}^*$, outputs σ^* on a chosen messages m^* where $\sigma^* = (c^*, R^*, S^*)$.

(U4) C executes unsigncryption algorithm with input as $(\sigma^*, ID_S{}^*, ID_j{}^*)$. If C outputs $m = m^*$, $\mathcal{A}_1$ wins the game.

Suppose $\mathcal{A}_1$ win the **Game EUF-CMA**, then $S^* = r_S\left(d_S h_4^S + x_S h_4^j\right) + d_S H^* + x_S J^*$, $H^* = h_2^{S*}\left(m^*, c^*, R^*, Y, Q_S, Q_j\right)$, $J^* = h_3^{S*}\left(m^*, c^*, R^*, Y, P_S, P_j\right)$, $R^* = r_S\left(d_S h_4^S + x_S h_4^j\right)P$. Based on the forking lemma[24], $\mathcal{A}_1$ can get another valid signcryption $(\sigma^{**}, {ID_S}^*, {ID_j}^*)$ on m^* according to replay attack with rearrangement in L_2 and L_3 . According to the birthday paradox, C may return two different hash values associated with the same input when answering a Hash query. Such successful birthday attack occurs with the probability of $\Pr[\text{forking\_}H_2\,] = \frac{1}{\sqrt{q_2}}$ for H_2 query and $\Pr[\text{forking\_}H_3\,] = \frac{1}{\sqrt{q_3}}$ for H_3 query.

We get $\sigma^{**} = (c^*, R^*, S^{**}), S^{**} = r_S\left(d_S h_4^S + x_S h_4^j\right) + d_S H^{**} + x_S J^{**}$

$$S^* P = R^* + d_S H^* P + x_S J^* P$$

$$S^{**} P = R^* + d_S H^{**} P + x_S J^{**} P$$

According to above expressions, we get $A = d_S = \frac{S^{**} - S^* + x_S J^* - x_S J^{**}}{H^{**} - H^*}$, $(H^{**} \neq H^*)$, from which, we conclude that Γ may solve the DL problem if successful birthday attack on H_2 occurs. Failure birthday attack on H_3 will generate $J^{**} = J^*$, which does not affect the solution of the DL problem. The running time to compute d_S is $t_{ds} \approx 3T_{msz}$, in which T_{msz} is the time for one scalar multiplication operation over Z_q^*.

With the above description, Γ wins to solve the DL problem only if when choosing ${ID_S}^*$ for challenge, successful birthday attack on H_2 occurs and the game is completed. But, Γ will terminate the game when any of the events $EU_1, EU_2, EU_3, EU_4, EU_5$ occurs.

EU_1: $\mathcal{A}_1$ does not choose $({ID_S}^*, *)$ for challeng, $\Pr\,[EU_1] = \frac{C_{q_c-1}^1}{C_{q_c}^1}$.

EU_2: $\mathcal{A}_1$ asks Rd_i query with ${ID_S}^*$,

$$\Pr\,[EU_2] = \frac{1}{C_{q_c}^1} + \frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \frac{1}{C_{q_c}^1} + \cdots + \underbrace{\frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \ldots \cdot \frac{C_{q_c-1}^1}{C_{q_c}^1}}_{q_d - 1} \cdot \frac{1}{C_{q_c}^1} = 1 - (\frac{q_c - 1}{q_c})^{q_d}.$$

EU_3: $\mathcal{A}_1$ asks Rsk_i query with ${ID_S}^*$,

$$\Pr[EU_3] = \frac{1}{C_{q_c}^1} + \frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \frac{1}{C_{q_c}^1} + \cdots + \underbrace{\frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \ldots \cdot \frac{C_{q_c-1}^1}{C_{q_c}^1}}_{q_{sk} - 1} \cdot \frac{1}{C_{q_c}^1} = 1 - (\frac{q_c - 1}{q_c})^{q_{sk}}.$$

EU_4: $\mathcal{A}_1$ asks R$_{sc}$ query with $\left(m^*, {ID_S}^*, {ID_j}^*\right)$,

$$\Pr[EU_4] = \left[\frac{1}{2^k} + \left(1 - \frac{1}{2^k}\right) \cdot \frac{1}{2^k} + \cdots + \underbrace{\left(1 - \frac{1}{2^k}\right) \cdot \left(1 - \frac{1}{2^k}\right) \cdot \ldots \cdot \left(1 - \frac{1}{2^k}\right)}_{q_{sc} - 1} \cdot \frac{1}{2^k}\right] = 1 - \left(\frac{2^k - 1}{2^k}\right)^{q_{sc}}$$

EU_5: $\mathcal{A}_1$ fails to use oracle H_2 and replay technique to generate one more valid ciphertext. $\Pr[EU_5] = 1 - \Pr[\text{forking\_}H_2\,]$.

Then, $\mathcal{A}_1$ will win the **Game EUF-CMA** with advantage ε and running time t, then an algorithm Γ can be constructed to solve the DL problem with advantage ε' and running time τ by interacting with $\mathcal{A}_1$.

$$\varepsilon' = (1 - \Pr\,[EU_1]) \cdot (1 - \Pr\,[EU_2]) \cdot (1 - \Pr\,[EU_3]) \cdot (1 - \Pr\,[EU_4]) \cdot (1 - \Pr\,[EU_5]) \cdot \varepsilon$$

$$= \frac{1}{q_c} \cdot (\frac{q_c - 1}{q_c})^{q_d} \cdot (\frac{q_c - 1}{q_c})^{q_{sk}} \cdot \left(\frac{2^k - 1}{2^k}\right)^{q_{sc}} \cdot \frac{1}{\sqrt{q_2}} \cdot \varepsilon$$

$$\tau \leq \textstyle\sum_{i=0}^4 q_i\, t_i + q_c\, t_c + q_d\, t_d + q_x\, t_x + q_{pk}\, t_{pk} + q_r\, t_r + q_{sc}\, t_{sc} + q_{usc}\, t_{usc} + 2t + t_{ds}$$

Lemma 4.

Suppose $\mathcal{A}_2$ win the **Game EUF-CMA** with advantage ε and running time t, then an algorithm Γ can be constructed to solve the instance of the DL problem in **Lemma 3** with advantage $\varepsilon'$and running time τ by interacting with $\mathcal{A}_2$.

$$\varepsilon' = \frac{1}{q_c} \cdot (\frac{q_c-1}{q_c})^{q_{sk}} \cdot \left(\frac{2^k-1}{2^k}\right)^{q_{sc}} \cdot (\frac{q_c-1}{q_c})^{q_x} \cdot (\frac{q_c-1}{q_c})^{q_{pk}} \cdot \frac{1}{\sqrt{q_3}} \cdot \varepsilon$$

$$\tau \leq \sum_{i=0}^{4} q_i\, t_i + q_c\, t_c + q_d\, t_d + q_x\, t_x + q_{pk}\, t_{pk} + q_r\, t_r + q_{sc}\, t_{sc} + q_{usc}\, t_{usc} + 2t + t_{xs}$$

Proof. To interact with $\mathcal{A}_2$, algorithm Γ runs the following steps to solve the instance of the DL problem .

(U1) Γ executes the SETUP algorithm and sends system params and master secret key s to $\mathcal{A}_2$.

(U2) Suppose that Γ will choose accepted sender S with identity $ID_S{}^*$ and a user $ID_j{}^*$for challenge. $\mathcal{A}_2$ asks the Γ for a polynomial number of the queries as shown in **Lemma 2**. Queries only contains conditions of $i \neq S$ and $i = S$, where receiver R should not be considered and specified.

(U3)(U4) steps are the same as that in **Lemma 3**.

$\mathcal{A}_2$ will get $A = x_S = \frac{S^{**}-S^{*}+d_S H^{*}-d_S H^{**}}{J^{**}-J^{*}}$, $J^{**} \neq J^{*}$, from which, we conclude that Γ may solve the DL problem if successful birthday attack on H_3 occurs. Failure birthday attack on H_2 will generate $H^{**} = H^{*}$, which does not affect the solution of the DL problem. The running time to compute x_S is $t_{xs} \approx 3T_{msz}$.

Γ will terminate the game when any of the events EU_1, EU_3, EU_4 , EU_6, EU_7 , EU_8 occurs.

EU_6: $\mathcal{A}_2$ asks Rx_i query with $ID_S{}^*$,

$$\Pr[EU_6] = \frac{1}{C_{q_c}^1} + \frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \frac{1}{C_{q_c}^1} + \cdots + \underbrace{\frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \ldots \cdot \frac{C_{q_c-1}^1}{C_{q_c}^1}}_{q_x-1} \cdot \frac{1}{C_{q_c}^1} = 1 - (\frac{q_c-1}{q_c})^{q_x}$$

EU_7: $\mathcal{A}_2$ asks Rpk_i query with $ID_S{}^*$,

$$\Pr[EU_7] = \frac{1}{C_{q_c}^1} + \frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \frac{1}{C_{q_c}^1} + \cdots + \underbrace{\frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \frac{C_{q_c-1}^1}{C_{q_c}^1} \cdot \ldots \cdot \frac{C_{q_c-1}^1}{C_{q_c}^1}}_{q_{pk}-1} \cdot \frac{1}{C_{q_c}^1} = 1 - (\frac{q_c-1}{q_c})^{q_{pk}}$$

EU_8: $\mathcal{A}_2$ fails to use oracle H_3 and replay technique to generate one more valid ciphertext. $\Pr[EU_8] = 1 - \Pr[\text{forking\_}H_3\,]$.

Similarly, $\mathcal{A}_2$ will win the **Game EUF-CMA** with advantage ε and running time t, then an algorithm Γ can be constructed to solve the DL problem with advantage ε' and running time τ by interacting with $\mathcal{A}_2$.

$$\varepsilon' = (1 - \Pr[EU_1]) \cdot (1 - \Pr[EU_3]) \cdot (1 - \Pr[EU_4]) \cdot (1 - \Pr[EU_6]) \cdot (1 - \Pr[EU_7]) \cdot (1 - \Pr[EU_8]) \cdot \varepsilon$$

$$= \frac{1}{q_c} \cdot (\frac{q_c-1}{q_c})^{q_{sk}} \cdot \left(\frac{2^k-1}{2^k}\right)^{q_{sc}} \cdot (\frac{q_c-1}{q_c})^{q_x} \cdot (\frac{q_c-1}{q_c})^{q_{pk}} \cdot \frac{1}{\sqrt{q_3}} \cdot \varepsilon$$

$$\tau \leq \sum_{i=0}^{4} q_i\, t_i + q_c\, t_c + q_d\, t_d + q_x\, t_x + q_{pk}\, t_{pk} + q_r\, t_r + q_{sc}\, t_{sc} + q_{usc}\, t_{usc} + 2t + t_{xs}$$

According to **Lemma 3** and **Lemma 4**, if $\mathcal{A}$ win the **Game EUF-CMA** in polynomial time, Γ can solve the DL problem, which is contradictory with the security assumption of DL problem. Then, we conclude that $\mathcal{A}$ cannot win the **Game EUF-CMA** and $EUF\_Adv_A(k)$ is negligible. Therefore, our scheme can provide unforgeability under the DL problem.

In light of the proof above, our proposed scheme can also resist the PKR attack, MPK attack and ESL attack.

5.2. Efficiency analysis

In this section, we evaluates our proposed scheme compared with other related ones. **Table 3** lists the computation time cost for referred cryptographic operations from research works [25, 26] and the lengths of parameters. Besides, time for hash and xor operations are trivial and can be neglected in the comparison. **Table 4** shows the efficiency of our scheme compared with related ones. Symbol √ denotes that the scheme supports the corresponding character while × denotes not.

Table 3. Notation in efficiency

Notation	Description	cost	Notation	Description
T_{mul}	One scalar multiplication operation over elliptic curve	2.21ms	$\|m\|$	length of message m
T_p	One pairing operation in group	20.04ms	$\|P\|$	the size of an element in G
T_{exp}	modular exponentiation in a cyclic group	5.31ms	$\|r\|$	the size of an element in finite field Z_q^*
T_m	Modular multiplication	$1T_{mul}{\approx}1200\ T_m$		

Table 4. Efficiency comparison among related CLSC scheme

Scheme	Efficiency			Supported features				
	Communication cost	Computation cost		Unforgeability	Confidentiality	ESL attack	PKR attack	MPK attack
		signcryption	unsigncryption					
[10]	$\|m\|+\|P\|+\|r\|$	$Tp + 5T_{mul}$	$5\ Tp + 2\ T_{mul}$	√	√	√	√	√
[11]	$\|P\|+\|r\|$	$Tp + T_{exp} + 2T_{mul}$	$5Tp + 2T_{mul}$	×	×	×	×	×
[12]	$\|m\|+2\|P\|$	$Tp + T_{exp} + 4T_{mul}$	$5Tp + T_{mul}$	×	×	×	×	×
[13]	$\|m\|+\|P\|+2\|r\|$	$Tp + 3T_{mul}$	$3Tp + 3T_{mul}$	×	×	×	×	×
[16]	$\|m\|+4\|r\|$	$5T_{exp}$	$7T_{exp}$	×	×	×	×	×
[22]	$\|m\|+2\|r\|$	$3\mathrm{T}_{\mathrm{exp}}$	$5\mathrm{T}_{\mathrm{exp}}$	×	×	×	×	×
ours	$\|m\|+\|P\|+\|r\|$	$3T_{mul} + 4\ T_m$	$5T_{mul}$	√	√	√	√	√

6. Conclusions

In this paper, we demonstrate the security weakness of several existing CLSC schemes, and present a CLSC scheme without pairing based on elliptic curve cryptosystem (ECC). Security proof shows that our scheme is secure to provide confidentiality and unforgeability resting on Gap Diffie-Hellman (GDH) assumption and discrete logarithm problem in the random oracle model. Compared with related CLSC schemes, the security and efficiency analysis show that our scheme satisfies more security characters with lowest time cost and slight higher communication cost.

References

[1] L. M. Kohnfelder, "Towards a practical public-key cryptosystem," *B.S. Thesis in Massachusetts Institute of Technology*, 1978. Article (CrossRef Link)

[2] A. Shamir, "Identity-based cryptosystems and signature schemes," *Lecture Notes in Computer Science*, vol.196, pp. 47-53, 1985. Article (CrossRef Link)

[3] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," *Lecture Notes in Computer Science*, vol.2894, pp. 452-473, 2003. Article (CrossRef Link)

[4] Y. L. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) plus cost(encryption)," *Advances in cryptology - crypto'97*, pp. 165-179, 1997.

Article (CrossRef Link)

[5] M. Barbosa and P. Farshim, "Certificateless signcryption,", in *proc. of ACM Symposium on Information, Computer and Communications Security*, pp. 369-372, March 20, 2008. Article (CrossRef Link)

[6] D. Aranha, R. Castro, J. López and R. Dahab, "Efficient certificateless signcryption," in *Proc. of 8th Brazilian Symposium on Information and Computer Systems Security*, 2008. Article (CrossRef Link)

[7] C. H. Wu and Z. X. Chen, "A new efficient certificateless signcryption scheme," in *Proc. of 2008 International Symposium on Information Science and Engineering*, pp. 661-664, December 20, 2008. Article (CrossRef Link)

[8] W. J. Xie and Z. Zhang, "Efficient and provably secure certificateless signcryption from bilinear maps," in *Proc. of 2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, pp. 558-562, June 25 - 27, 2010. Article (CrossRef Link)

[9] Z. H. Liu, Y. P. Hu, X. S. Zhang and H. Ma, "Certificateless signcryption scheme in the standard model," *Information Sciences*, vol.180, no.3, pp. 452-464, February, 2010. Article (CrossRef Link)

[10] S. K. H. Islam and F. Li, "Leakage-free and provably secure certificateless signcryption scheme using bilinear pairings," *Computer Journal*, vol.58, no.10, pp. 2636-2648, October, 2015. Article (CrossRef Link)

[11] F. Li, M. Shirase and T. Takagi, " Certificateless hybrid signcryption," *Mathematical and Computer Modelling*, vol. 57, no.3-4, pp. 324-343, 2013. Article (CrossRef Link)

[12] C. Zhou, W. Zhou and X. Dong, "Provable certificateless generalized signcryption scheme," *Designs Codes and Cryptography*, vol.71, no.2, pp. 331-346, May, 2014. Article (CrossRef Link)

[13] A. Yin and H. Liang, "On security of a certificateless hybrid signcryption scheme," *Wireless Personal Communications*, vol.85, no.4, pp. 1727-1739, December, 2015. Article (CrossRef Link)

[14] M. H. Au, J. Chen, J. K. Liu, Y. Mu, D. S. Wong and G. Yang, "Malicious kgc attacks in certificateless cryptography," in *Proc. of 2nd ACM Symposium on Information, Computer and Communications Security*, pp. 302-311, March 20 -22, 2007. Article (CrossRef Link)

[15] J. Weng, G. X. Yao, R. H. Deng, M. R. Chen and X. X. Li, "Cryptanalysis of a certificateless signcryption scheme in the standard model," *Information Sciences*, vol.181, no.3, pp. 661-667, February, 2011. Article (CrossRef Link)

[16] S. S. D. Selvi, S. S. Vivek and C. P. Rangan, "Cryptanalysis of certificateless signcryption schemes and an efficient construction without pairing," *Lecture Notes in Computer Science*, vol.6151, pp. 75-92, 2010. Article (CrossRef Link)

[17] S. K. H. Islam, "A provably secure id-based mutual authentication and key agreement scheme for mobile multi-server environment without esl attack," *Wireless Personal Communications*, vol.79, no.3, pp. 1975-1991, December, 2014. Article (CrossRef Link)

[18] H. Li, H. Zhu and Y. M. Wang, "Certificateless signcryption scheme without pairing," *Computer Research and Development*, vol.47, no. 9, pp. 1587-1594, 2010. Article (CrossRef Link)

[19] W. Liu and C. Xu, "Certificateless signcryption scheme without bilinear pairing," *Journal of Software*, vol.22, no.8, pp. 1918-1926, 2011. Article (CrossRef Link)

[20] X. Jing, " Provably secure certificateless signcryption scheme without pairing," in *Proc. of 2011 International Conference on Electronic and Mechanical Engineering and Information Technology*, pp. 4753-4756, August 12- 14, 2011. Article (CrossRef Link)

[21] D. He, "Security analysis of a certificateless signcryption scheme," *Journal of Software*, vol.24, no.3, pp. 618-622, 2013. Article (CrossRef Link)

[22] W. B. Shi, N. Kumar, P. Gong and Z. Z. Zhang, "Cryptanalysis and improvement of a certificateless signcryption scheme without bilinear pairing," *Frontiers of Computer Science*, vol.8, no.4, pp. 656-666, August, 2014. Article (CrossRef Link)

[23] Y. Lu and J. Li, "Provably secure certificate-based signcryption scheme without pairings," *Ksii Transactions on Internet and Information Systems*, vol.8, no.7, pp. 2554-2571, July, 2014. Article (CrossRef Link)

[24] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Proc. of Advances in cryptology - eurocrypt '96*, pp. 387-398, Springer, Berlin, 1996. Article (CrossRef Link)

[25] D. He, J. Chen and J. Hu, "An id-based proxy signature schemes without bilinear pairings," *Annals of Telecommunications-Annales Des Telecommunications*, vol.66, no.11-12, pp. 657-662, December, 2011. Article (CrossRef Link)

[26] H. Arshad and M. Nikooghadam, "Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems," *Journal of Medical Systems*, vol.38, no.12, December, 2014. Article (CrossRef Link)

LiLing Cao: received her M.S. degree in Science and Technology of Electronic Information from Central South University in China in 2007, received her Ph.D. degree in Measurement and Control Technology and Automation Instrument from Tong Ji University in China in 2017. She had worked in Shanghai Ocean University for 10 years. Her research interests include security protocol and wireless communication.

WangCheng Ge: received his Ph.D. degree in the department of Electrical engineering and computer science from University of Siegen in German in 1998.Then, he did post-doctoral research work in Technical University of Munich. He had worked in Sino-German College in Tong Ji University for 13 years as the chair of Rhodes and Schwartz communication network project fund department.