Provably secure certificateless encryption scheme in the standard model

Lunzhi Deng[1*], Tian Xia[1], Xiuru He[1]

1. School of Mathematical Sciences, Guizhou Normal University, Guiyang 550001, China.

*Corresponding author: Lunzhi Deng

Received August 30, 2019; revised March 16, 2020; accepted March 31, 2020; published June 30, 2020

Abstract

Recently, numerous certificateless encryption (CLE) schemes have been introduced. The security proofs of most schemes are given under the random oracle model (ROM). In the standard model, the adversary is able to calculate the hash function instead of asking the challenger. Currently, there is only one scheme that was proved to be secure in SM. In this paper, we constructed a new CLE scheme and gave the security proofs in SM. In the new scheme, the size of the storage space required by the system is constant. The computation cost is lower than other CLE schemes due to it needs only two pairing operations.

Keywords: Certificateless encryption, Pairing, Standard model, Diffie-Hellman problem, Security

1. Introduction

With the continuous advancement of communication technique, a large amount of information is transmitted through the network, which improves work efficiency and brings convenience to people's lives. In the same way, this also leads criminals to easily steal information from others through the Internet. People enjoy the convenience brought by information technology and also bear the risk of disclosure of personal privacy information. Public key encryption technology has become an important means to achieve information security. In order to meet different needs, researchers have done much work to build specific public key encryption schemes in recent years.

In public key infrastructure (PKI), the user freely picks his/her own private key, then generates a public key and sends it to the certification authority (CA). CA generates a certificate to bind the user to his/her public key. A large amount of fees are used for the safekeeping, storage and transmission of certificates. To resolve the problem, Shamir [1] came up with identity-based cryptography. The user's sole personal information (email address, identity number, etc.) is his/her public key. Private key generator (PKG) yields the private key based on the public key and forwards it to the user. The information security of all users will be threatened if PKG is captured by an adversary. In 2003, Al-Riyami and Paterson [2] came up with certificateless cryptography. For one thing, the user picks a confidential value and yields a partial public key. For another, the user gets a partial private key, yielded by a key generation center (KGC) based on the identity information, through an authenticated channel.

1.1. Related work

Al-Riyami and Paterson [2] came up with the first CLE scheme. But, Libert and Quisquater [3] demonstrated that the scheme [2] is insecure, and put forward a means to construct CLE schemes with provably security. In 2010, Sun and Li [4] proposed a new CLE scheme with short-ciphertext, and proved it to be secure against chosen-ciphertext attacks (CCAs). In 2005, Baek et al. [5] presented a CLE scheme that does not require pairing operation. Sun et al. [6] indicated that the scheme [5] can achieve the security goals only in a weaker model, where Type I adversary is not allowed to change the user's public key. In 2013, Yan et al. [7] put forward a pairing-free CLE scheme and provided the security proofs in ROM. In same year, Guo et al. [8] brought forward a CLE scheme that does not require pairing operation. However, Deng et al. [9] pointed out that there are security flaws in scheme [8], then proposed a modified scheme. In 2018, Zhou et al. [10] came up with a CLE scheme that does not require pairing operation, and showed that it is secure against CCAs. In 2015, SK Hafizul et al. [11] put forward a certificateless multi-receiver encryption (CLMRE) scheme, and provided security proofs in ROM. In 2017, He et al. [12] proposed a pairing-free CLMRE scheme, which is efficient due to no Hash-to-Point (HTP) operation is required. In

the same year, Gao et al. [13] brought forward a new CLMRE scheme, and proved that the receiver's identity information will not be leaked.

In 2007, Huang and Wong [14] came up with a common structure of CLE, which is provably secure in SM against the KGC attacks. In 2008, Dent et al. [15] presented a new CLE scheme, and asserted that it achieved confidentiality of the message in SM. But, Hwang et al. [16] indicated that the ciphertext indistinguishability against the KGC attacks does not hold for the scheme [15], then constructed a new CLE scheme. In 2009, Zhang and Wang [17] pointed out that the ciphertext indistinguishability against the key replacement attacks does not hold for the scheme [16], then constructed a new CLE scheme. However, Shen et al. [18] indicated that the ciphertext indistinguishability against the type II adversary does not hold for the scheme [17]. In 2014, Cheng et al. [19] evidenced that the ciphertext indistinguishability against the KGC attacks does not hold for the scheme [16], then proposed an improved scheme with provably security in SM. Reza et al. [20] put forward a common means to design CLE schemes with provably security in SM against CCAs, which come from a secure identity-based encryption scheme against chosen-plaintext attacks (CPAs).

1.2. Motivations and contributions

To increase security levels and reduce computing costs, researchers have proposed many CLE schemes. However, two problems remain in these schemes.

• Security proofs for most known CLE schemes are given in ROM

As we all know, the cryptography scheme provided with the security proofs in the ROM may be unsafe in a real situation. Therefore, these CLE schemes with provable security in ROM may be insecure in actual scenarios.

• High computation and storage costs

In the last ten years, scholars have proposed several concrete CLE schemes [15, 16, 17, 19], and tried to prove that they are secure in SM. However, there is only one scheme [19] that has been proven to be secure in SM. In these schemes [15, 16, 19], the size of the storage space required by the system is linearly related to the size of the user's identity information, and the times of addition operations on the elliptic curve group increases linearly with the size of the user's identity information. These increase the storage burden and computation cost for the users and the key generation center.

It is attractive to design an efficient CLE scheme and provide the security proofs in SM. We summarized the contributions as follows.

• We introduce the system model and security requirements of a CLE scheme in SM.

• We bring forward a new CLE scheme and offer the security proofs in SM. In order to get the hash function value, the adversary does not need to query the challenger, but directly calculates the hash function.

• We give a comparison of the efficiency between three CLE schemes. In the new scheme, it was constant that the size of the storage space required by the system. It was constant that the number of three kinds of operations (addition, scalar multiplication, and pairing), so the computational cost is lower than other CLE schemes.

1.3. Organization

We introduce mathematical tools, system model and security requirements in Section 2, Section 3, and Section 4, respectively. We give a new CLE scheme and the security proofs in Section 5 and Section 6, respectively. We demonstrate an efficiency analysis of three CLE schemes in Section 7. We present some conclusions in Section 8.

2. Preliminaries

In this section, we introduce two mathematical tools: bilinear pairing and decisional bilinear Diffie-Hellman problem. **Table 1** lists the notations used in the paper.

Table 1. Notations

Symbol	Meaning
F_p	A prime finite field
q	A prime number
Z_q^*	A set of positive integers less than q.
$\hat{e}: G_1 \times G_1 \to G_2$	A bilinear pairing
P	A generator of the group G_1.
x, P_{pub}	The master secret key and public key of system, $x \in Z_q^*$ and $P_{pub} = xP$
H_1, H_2, H_3	Three secure hash functions.
ID_i	The identity of i^{th} user.
D_i	The partial private key of i^{th} user, where $D_i = (R_i, d_i)$, $R_i = r_i P$, $d_i = r_i + k_i x$, $k_i = H_1(ID_i, R_i)$

t_i	The secret value of i^{th} user and $T_i = t_i P$.
PK_i	The public key of i^{th} user, where $PK_i = (T_i, R_i)$.

Bilinear pairing

Let $\hat{e}: G_1 \times G_1 \to G_2$ be a mapping with the following attributes, where $G_1 = (P)$ and G_2 are the additive and multiplicative groups of the q order, respectively

- Bilinearity: $\hat{e}(aP_1, bP_2) = \hat{e}(P_1, P_2)^{ab}$ for all $P_1, P_2 \in G_1$ and $a, b \in Z_q^*$.
- Non-degeneracy: There exist $P_1, P_2 \in G_1$ such that $\hat{e}(P_1, P_2) \neq 1_{G_2}$.
- Computability: It is not difficult to compute $\hat{e}(P_1, P_2)$ for all $P_1, P_2 \in G_1$.

Definition 1. Decisional bilinear Diffie-Hellman (DBDH) problem. Let $\hat{e}: G_1 \times G_1 \to G_2$ be a bilinear pairing. For $P \in G_1$, $X \in G_2$, input a tuple (P, aP, bP, cP, X), decide whether $X = \hat{e}(P, P)^{abc}$.

3. System Model

A CLE scheme involves three distinct entities: key generation center (KGC), encryptor and decryptor, as shown in **Fig. 1**.

• KGC: It generates and publishes the system parameters. In addition, it yields a partial private key for the user.

• Encryptor: He encrypts a message to be a ciphertext by using the receiver's public key, then forwards that to the receiver.

• Decryptor: He obtains a message by decrypting the ciphertext with own private key.

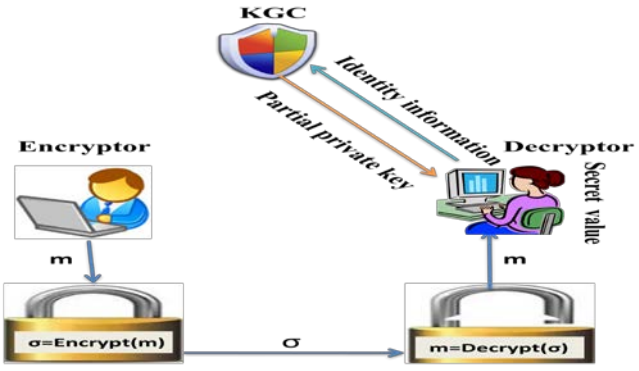

Fig. 1. Certificateless encryption

A CLE scheme is constituted with the following six algorithms:

- Setup: Inputs a parameter v, KGC yields the *msk* (master secret key) and the *params* (system parameters).
- PPK-Extract: Inputs an identity $ID_i \in \{0,1\}^*$, KGC yields a partial private key D_i and dispatches it to the user through a reliable channel..
- SV-Set: The user ID_i picks a secret value t_i.
- UPK-Generate: The user ID_i outputs his public key PK_i.
- Encrypt: Inputs a tuple (m, ID_i, PK_i), the encryptor outputs a ciphertext σ.
- Decrypt: Inputs a tuple (σ, ID_i, PK_i), the decryptor outputs a message m or the symbol "0".

4. Security Requirements

We described the security requirements in this section.

Definition 2. If the adversary's ascendency is insignificant in the coming two games, then the CLE scheme is indistinguishable (IND-CLE)

Game I. A challenger $\mathfrak{C}$ and a Type I adversary A_1 play this game together.

Initialization. $\mathfrak{C}$ gets *msk* and *params* by implementing the Setup algorithm, maintains *msk* secret and forwards *params* to A_1

Phase 1. A_1 performs multiple types of queries.

- UPK-Query: $\mathfrak{C}$ outputs a user public key PK_i when A_1 inputs an identity ID_i.
- UPK-Replacement: $\mathfrak{C}$ replaces PK_i with PK_i' when A_1 inputs a tuple (ID_i, PK_i'),
- PPK-Query: $\mathfrak{C}$ outputs a partial public key D_i when A_1 submits an identity ID_i. $\mathfrak{C}$ refuses to answer if the value R_i has been replaced.
- SV-Query: $\mathfrak{C}$ outputs a secret value t_i when A_1 inputs an identity ID_i. $\mathfrak{C}$ refuses to answer if the value T_i has been replaced.
- ENC-Query: $\mathfrak{C}$ outputs a ciphertext σ when A_1 submits a tuple (m, ID_i, PK_i).
- DEC-Query: $\mathfrak{C}$ returns a message m or the symbol "0" when A_1 submits a tuple (σ, ID_i, PK_i).

Challenge. A_1 submits a tuple (m_0, m_1, ID^*, PK^*), $\mathfrak{C}$ randomly selects a bit $\mu \in \{0,1\}$ and offers A_1 with $\sigma^* = \text{Encrypt}(m_\mu, ID^*, PK^*)$. That fulfills the following conditions:

1. m_0 and m_1 are two equal length messages.
2. A_1 did not make the PPK-Query for ID^*.

Phase 2. A_1 executes various queries again, which fulfills the following requirements.

1. A_1 did not make the PPK-Query for ID^*.
2. A_1 did not make the DEC-Query for σ^*.

Response. A_1 returns a bit μ' and wins if $\mu' = \mu$.

The advantage of A_1 is defined as: $Adv_{A_1}^{IND-CLE} = |\Pr[\mu' = \mu] - \frac{1}{2}|$

Game II. A challenger $\mathfrak{C}$ and a Type II adversary A_2 play this game together.

Initialization. $\mathfrak{C}$ gets msk and $params$ by implementing the Setup algorithm, then forwards them to A_2.

Phase 1. A_2 makes a series of queries as those in Game I.

Challenge. A_2 submits a tuple (m_0, m_1, ID^*, PK^*), $\mathfrak{C}$ randomly selects a bit $\mu \in \{0,1\}$, provides A_2 with $\sigma^* = \text{Encrypt}(m_\mu, ID^*, PK^*)$, which satisfy the following requirements.

1. m_0 and m_1 are two equal length messages.
2. A_2 did not perform SV-Query for ID^*.
3. A_2 did not perform UPK-Replacement for T^*.

Phase 2. A_2 executes various queries again, which satisfy the following requirements.

1. A_2 did not make SV-Query for ID^*.
2. A_2 did not perform UPK-Replacement for T^*.
3. A_2 did not make the DEC-Query for σ^*.

Response. A_2 returns a bit μ' and wins if $\mu' = \mu$.

The advantage of A_2 is defined as: $Adv_{A_2}^{IND-CLE} = |\Pr[\mu' = \mu] - \frac{1}{2}|$

5. New scheme

We constructed a new CLE scheme in this section. In the three schemes [15, 16, 19], the private key is generated based on each bit of the user's identity information. In our scheme, the identity information of the user is a whole, and the private key is generated based on the

identity information, rather than directly related to each bit of the identity information. Our scheme is constituted with the following algorithms.

- Setup: Inputs a security parameter v, KGC does as follows.

  1. Chooses two groups G_1 and G_2 with prime order $q > 2^v$, a generator P of G_1 and a bilinear pairing $\hat{e}: G_1 \times G_1 \to G_2$.

  2. Selects three hash functions $H_1, H_2 : \{0,1\}^* \to Z_q^*$, $H_3 : \{0,1\}^* \to \{0,1\}^{l_1+l_2}$.

  3. Sets the message space $M = \{0,1\}^{l_1}$.

  4. Chooses a number $x \in Z_q^*$, computes $P_{pub} = xP$ and sets $msk = \{x\}$.

  5. Publish $params = \{G_1, G_2, q, \hat{e}, P, P_{pub}, H_1, H_2, H_3\}$.

- PPK-Extract: Inputs an identity $ID_i \in \{0,1\}^*$, KGC randomly selects $r_i \in Z_q^*$ and computes $R_i = r_i P$, $k_i = H_1(ID_i, R_i)$, $d_i = r_i + k_i x$, then sends $D_i = (R_i, d_i)$ to the user through an authenticated channel.
- SV-Set: The user ID_i chooses at random $t_i \in Z_q^*$.
- UPK-Generate: The user ID_i computes $T_i = t_i P$, and sets $PK_i = (T_i, R_i)$.
- Encrypt: To transmit a message $m \in M$ to Bob (ID_B / PK_B), Alice carries out following steps:

  1. Selects at random $u, v \in Z_q^*$, and $w \in \{0,1\}^{l_2}$.

  2. Computes $U = uP, V = vP$, $k_B = H_1(ID_B, R_B)$, $h = H_2(m, w, U, V, ID_B, PK_B)$,

  $$E = \hat{e}(uvP, R_B + k_B P_{pub} + hT_B) \text{ and } C = H_3(U, V, E, h) \oplus m \| w.$$

  3. Outputs the tuple $\sigma = (C, U, V, h)$.

- Decrypt: On receive the tuple $\sigma = (C, U, V, h)$, Bob carries out following steps:.

  1. Computes $E = \hat{e}(U, V)^{d_B + ht_B}$, $m \| w = H_3(U, V, E, h) \oplus C$.

2. Checks whether $H_2(m,w,U,V,ID_B,PK_B)=h$. Accepts the message m if the equality holds. Otherwise, rejects.

- On correctness

$$\hat{e}(U,V)^{d_B+ht_B}=\hat{e}\left((d_B+ht_B)U,V\right)=\hat{e}\left((d_B+ht_B)uP,vP\right)$$

$$=\hat{e}\left(uvP,(r_B+k_Bx+ht_B)P\right)=\hat{e}\left(uvP,R_B+k_BP_{pub}+hT_B\right)=E$$

It is clear that the receiver can get the message by calling the decryption algorithm.

6. Security of scheme

We will give the security proofs in SM in this section. To obtain the hash function value, the adversary does not need to query the challenger, but directly calculates the hash function.

Theorem 1. Our scheme is indistinguishable against the Type I adversary in SM if the DBDH problem is hard.

Proof. Suppose that the tuple (P,aP,bP,cP,X) is an example of DBDH problem. In order to determine whether $X=\hat{e}(P,P)^{abc}$, $\mathfrak{C}$ will act as the challenger.

Initialization. Executing the Setup algorithm, $\mathfrak{C}$ gets

$$params=\{G_1,G_2,q,\hat{e},P,P_{pub}=xP,H_1,H_2,H_3\} \text{ and } msk=\{x\}$$

then sends the *params* to A_1.

Phase 1. Prior to other queries, an identity is first used for the public key queries. In order to store the query and answer, several initially empty lists are set.

- UPK-Query: $\mathfrak{C}$ maintains a list L_U of tuple (ID_i,t_i,r_i). When A_1 submits an identity ID_i, $\mathfrak{C}$ does as follows.

  Case 1. At the j^{th} query, picks at random $t_j\in Z_q^*$, sets $ID_j=ID^{\diamond}$, $PK_j=PK^{\diamond}=(t_jP,aP)$, then adds the tuple $(ID_j,t_j,\diamond)$ to the list L_U.

  Case 2. For $i\neq j$, $\mathfrak{C}$ picks at random $t_i,r_i\in Z_q^*$ and returns $PK_i=(t_iP,r_iP)$, then adds the tuple (ID_i,t_i,r_i) to the list L_U.

- UPK-Replace: $\mathfrak{C}$ maintains a list L_R of tuple $\left(ID_i, PK_i, PK_i^{'}\right)$. When A_1 submits a tuple $\left(ID_i, PK_i^{'}\right)$. $\mathfrak{C}$ replaces PK_i with $PK_i^{'}$ and adds $\left(ID_i, PK_i, PK_i^{'}\right)$ to the list L_R.
- PPK-Query: $\mathfrak{C}$ maintains a list L_D of tuple $\left(ID_i, D_i\right)$. When A_1 submits an identity ID_i. If $ID_i = ID^{\Diamond}$, $\mathfrak{C}$ fails and stops. Otherwise, $\mathfrak{C}$ finds $\left(ID_i, t_i, r_i\right)$ in the list L_U, outputs the D_i by executing the PPK-Extract algorithm, then adds $\left(ID_i, D_i\right)$ to the list L_D.
- SV-Query: When A_1 submits an identity ID_i. $\mathfrak{C}$ finds $\left(ID_i, t_i, r_i\right)$ in the list L_U, returns t_i.
- ENC-Query: When A_1 inputs a tuple $\left(m, ID_i, PK_i\right)$, $\mathfrak{C}$ executes the Encrypt algorithm and returns a ciphertext σ.
- DEC-Query: When A_1 submits a tuple $\left(\sigma, ID_i, PK_i\right)$, $\mathfrak{C}$ carries out following steps.
  1. $ID_i \neq ID^{\Diamond}$ and $ID_i \notin L_R$, $\mathfrak{C}$ returns a message m by calling the Decrypt algorithm.
  2. $ID_i \in L_R$, then the $PK_i = \left(r_i P, t_i P\right)$ has been updated to $PK_i^{'} = \left(r_i^{'} P, t_i^{'} P\right)$. If $r_i^{'} \neq r_i$ (or $t_i^{'} \neq t_i$), A_1 must send $r_i^{'}$ (or $t_i^{'}$) to $\mathfrak{C}$, $\mathfrak{C}$ executes the Decrypt algorithm and returns a message m.
  3. $ID_i = ID^{\Diamond}$, $\mathfrak{C}$ fails.

Challenge. A_1 submits a tuple $\left(m_0, m_1, ID^*, PK^*\right)$ satisfying the requirements in the Game I. If $ID^* \neq ID^{\Diamond}$, $\mathfrak{C}$ aborts. Otherwise $ID^* = ID^{\Diamond} = ID_j$, $\mathfrak{C}$ randomly selects a bit $\mu \in \{0,1\}$ and does as follows.

1. Finds $\left(ID_j, t_j, \Diamond\right)$ in list L_U and computes $k_j = H_1\left(ID_J, aP\right)$.
2. Picks at random $w \in \{0,1\}^{l_2}$, computes $h = H_2\left(m_\mu, w, bP, cP, ID^*, t_j P, aP\right)$.
3. Computes $E = X \cdot \hat{e}\left(bP, cP\right)^{k_j x + h t_j}$.
4. Computes $C = H_3\left(bP, cP, E, h\right) \oplus m_\mu \,||\, w$.

5. Outputs the ciphertext $\sigma^* = (C, bP, cP, h)$.

Phase 2. A_1 executes various queries again, which satisfy the terms in the Game I.

Response. A_1 outputs a bit $\mu' \in \{0,1\}$.

Solve DBDH problem. $\mathfrak{C}$ returns "1" if $\mu' = \mu$. Otherwise, returns "0". If $X = \hat{e}(P,P)^{abc}$, then

$$E = X \cdot \hat{e}(bP, cP)^{k_j x + ht_j} = \hat{e}(P,P)^{abc} \cdot \hat{e}\left(\left(k_j x + ht_j\right) bP, cP\right)$$

$$= \hat{e}\left(\left(a + k_j x + ht_j\right) bP, cP\right) = \hat{e}\left(bcP, \left(a + k_j x + ht_j\right) P\right)$$

$$= \hat{e}\left(bcP, R_j + k_j P_{pub} + hT_j\right)$$

Therefore, σ^* is a valid ciphertext. Since A_1 has advantage ε. So

$$\Pr[\mathfrak{C} \to 1 \mid X = \hat{e}(P,P)^{abc}] = \Pr[\mu' = \mu \mid X = \hat{e}(P,P)^{abc}] = \frac{1}{2} + \varepsilon.$$

If $X \neq \hat{e}(P,P)^{abc}$, then σ^* is an invalid ciphertext. Each part in σ^* has the same distribution for μ=0 and $\mu = 1$.So A_1 has no superiority to differentiate the bit μ. Hence

$$\Pr[\mathfrak{C} \to 1 \mid X \neq \hat{e}(P,P)^{abc}] = \Pr[\mu' = \mu \mid X \neq \hat{e}(P,P)^{abc}] = \frac{1}{2}.$$

Probability. Let q_U, q_R, q_K and q_D be the number of UPK-Query, UPK-Replace, PPK-Query, and DEC-Query, respectively. Three events are indicated as follows.

π_1: A_1 did not perform UPK-Replacement for $R^{\Diamond}$ and did not make the PPK-Query for $ID^{\Diamond}$.

π_2: $\mathfrak{C}$ did not fail in decryption queries.

π_3: $ID^* = ID^{\Diamond}$.

Obtaining the following results is not difficult.

$$\Pr[\pi_1] = \frac{q_U - q_R - q_K}{q_U}. \Pr[\pi_2 \mid \pi_1] = \left(1 - \frac{1}{q_U}\right)^{q_D} \approx e^{-\frac{q_D}{q_U}}. \Pr[\pi_3 \mid \pi_1 \wedge \pi_2] = \frac{1}{q_U - q_R - q_K}.$$

$$\Pr[\mathfrak{C}\ success] = \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] = \Pr[\pi_1] \bullet \Pr[\pi_2 \mid \pi_1] \bullet \Pr[\pi_3 \mid \pi_1 \wedge \pi_2]$$

$$= \frac{q_U - q_R - q_K}{q_U} \bullet e^{-\frac{q_D}{q_U}} \bullet \frac{1}{q_U - q_R - q_K} \approx \frac{1}{q_U} e^{-\frac{q_D}{q_U}}$$

Therefore, if A_1 can win with the probability ε in the Game I, then $\mathfrak{C}$ is able to decide whether $X = \hat{e}(P,P)^{abc}$ with the probability $\frac{\varepsilon}{q_U} e^{-\frac{q_D}{q_U}}$.

Theorem 2. Our scheme is indistinguishable against the Type II adversary in SM if the DBDH problem is hard .

Proof. Suppose that the tuple (P, aP, bP, cP, X) is an example of DBDH problem. In order to determine whether $X = \hat{e}(P,P)^{abc}$, $\mathfrak{C}$ will play the role of challenger..

Initialization. Executing the Setup algorithm, $\mathfrak{C}$ obtains

$$params = \{G_1, G_2, q, \hat{e}, P, P_{pub} = xP, H_1, H_2, H_3\} \text{ and } msk = \{x\}$$

and forwards them to A_2.

Phase 1. Prior to other queries, an identity is first used for public key queries. In order to store the query and answer, several initially empty lists are set.

- UPK-Query: $\mathfrak{C}$ safeguards a list L_U of tuple (ID_i, t_i, r_i). When A_2 inputs an identity ID_i, $\mathfrak{C}$ does as follows.

Case 1. At the j^{th} query, picks at random $r_j \in Z_q^*$, sets $ID_j = ID^{\Diamond}$ and $PK_j = PK^{\Diamond} = (aP, r_j P)$, then puts the tuple $(ID_j, \Diamond, r_j)$ in the list L_U.

Case 2. For $i \neq j$, $\mathfrak{C}$ randomly selects $t_i, r_i \in Z_q^*$ and returns $PK_i = (t_i P, r_i P)$, then puts the tuple (ID_i, t_i, r_i) in the list L_U.

- UPK-Replacement: Same as that in Theorem 1.
- PPK-Query: $\mathfrak{C}$ safeguards a list L_D of tuple (ID_i, D_i). When A_2 submits an identity ID_i, $\mathfrak{C}$ finds (ID_i, t_i, r_i) in list L_U, gives the D_i by executing the PPK-Extract algorithm, then adds (ID_i, D_i) to list L_D.
- SV-Query: When A_2 submits an identity ID_i. If $ID_i = ID^{\Diamond}$, $\mathfrak{C}$ terminates the game. Otherwise, $\mathfrak{C}$ finds (ID_i, t_i, r_i) in the list L_U, and returns t_i.
- ENC-Query, DEC-Query: Same as that in Theorem 1.

Challenge. A_2 inputs a tuple (m_0, m_1, ID^*, PK^*) satisfying the requirements in the Game II. If $ID^* \neq ID^{\Diamond}$, $\mathfrak{C}$ aborts. Otherwise $ID^* = ID^{\Diamond} = ID_j$, $\mathfrak{C}$ picks a bit $\mu \in \{0,1\}$ and does as follows:

1. Finds $(ID_j, \Diamond, r_j)$ in list L_U and computes $k_j = H_1(ID_j, r_j P)$.

2. Randomly selects $w \in \{0,1\}^{l_2}$, computes $h = H_2(m_\mu, w, bP, cP, ID^*, aP, r_j P)$.

3. Computes $E = X^h \cdot \hat{e}(bP, cP)^{k_j x + r_j}$.

4. Computes $C = H_3(bP, cP, E, h) \oplus m_\mu \| w$.

5. Returns the ciphertext $\sigma^* = (C, bP, cP, h)$.

Phase 2. A_2 executes various queries again, which satisfy the terms in the Game II.

Response. A_2 outputs a bit $\mu' \in \{0,1\}$.

Solve DBDH problem. $\mathfrak{C}$ outputs "1" if $\mu' = \mu$. Otherwise, outputs "0". If $X = \hat{e}(P,P)^{abc}$, then

$$E = X^h \cdot \hat{e}(bP, cP)^{k_j x + r_j} = \hat{e}(P,P)^{habc} \cdot \hat{e}((k_j x + r_j) bP, cP)$$

$$= \hat{e}((ha + k_j x + r_j) bP, cP) = \hat{e}(bcP, (ha + k_j x + r_j) P)$$

$$= \hat{e}(bcP, R_j + k_j P_{pub} + hT_j)$$

Therefore, σ^* is a valid ciphertext. Since A_2 has advantage ε. So

$$\Pr[\mathfrak{C} \to 1 \mid X = \hat{e}(P,P)^{abc}] = \Pr[\mu' = \mu \mid X = \hat{e}(P,P)^{abc}] = \frac{1}{2} + \varepsilon .$$

If $X \neq \hat{e}(P,P)^{abc}$, then σ^* is a invalid ciphertext. Each part in σ^* has the same distribution for $\mu = 0$ and $\mu = 1$. So A_2 has no superiority to differentiate the bit μ. Hence

$$\Pr[\mathfrak{C} \to 1 \mid X \neq \hat{e}(P,P)^{abc}] = \Pr[\mu' = \mu \mid X \neq \hat{e}(P,P)^{abc}] = \frac{1}{2}.$$

Probability. Let q_U, q_R, q_S and q_D be the number of UPK-Query, UPK-Replace, SV-Query, and DEC-Query, respectively. Three events are indicated as follows.

π_1: A_2 did not replace the value $T^{\Diamond}$ and did not make the SV-Query for $ID^{\Diamond}$.

π_2: $\mathfrak{C}$ does not fail in decryption queries.

$\pi_3 : ID^* = ID^{\diamond}$.

It is not difficult to obtain the following results.

$$\Pr[\pi_1] = \frac{q_U - q_R - q_S}{q_U} . \Pr[\pi_2 \mid \pi_1] = \left(1 - \frac{1}{q_U}\right)^{q_D} \approx e^{-\frac{q_D}{q_U}} . \Pr[\pi_3 \mid \pi_1 \wedge \pi_2] = \frac{1}{q_U - q_R - q_S} .$$

$$\Pr[\mathfrak{C}\ success] = \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] = \Pr[\pi_1] \bullet \Pr[\pi_2 \mid \pi_1] \bullet \Pr[\pi_3 \mid \pi_1 \wedge \pi_2]$$

$$= \frac{q_U - q_R - q_S}{q_U} \bullet e^{-\frac{q_D}{q_U}} \bullet \frac{1}{q_U - q_R - q_S} \approx \frac{1}{q_U} e^{-\frac{q_D}{q_U}}$$

Therefore, if A_2 can win with the probability ε in the Game II, then $\mathfrak{C}$ is able to decide whether $X = \hat{e}(P,P)^{abc}$ with the probability $\frac{\varepsilon}{q_U} e^{-\frac{q_D}{q_U}}$

7. Efficiency and comparison

We give a contrast on the efficiency of the three CLE schemes in this section. **Table 2** lists several notations used in this section.

Table 2. Notations

Symbol	Meaning
B_P	A bilinear pairing operation.
S_{G_1}	A scale multiplication operation in G_1.
A_{G_1}	A addition operation in G_1.
M_{G_2}	A multiplication operation in G_2.
E_{G_2}	An exponentiation operation in G_2.
H	A general hash operation.
B_{ID}	The bit string of user's identity information, where $\lvert B_{ID} \rvert = n$

$B[i]_{ID}$	The i^{th} bit of user's identity information.
Δ_{ID}	The set of indices i such that $B[i]_{ID}=1$, namely $\Delta_{ID}=\{i: B[i]_{ID}=1\}$.
$\lvert G_1\rvert$	An element in G_1.
$\lvert G_2\rvert$	An element in G_2.
$\lvert Z_q^*\rvert$	An element in Z_q^*.

Table 3. Running time (in milliseconds)

B_p	S_{G_1}	A_{G_1}	M_{G_2}	E_{G_2}	H
5.427	2.165	0.013	0.001	0.339	0.007

For fairness and reasonableness, we analyze the three CLE schemes by using the third-party data. Implementing the basic cryptographic operations on a computer (with the Window 8 operating system, 4G bytes memory and an I5-4460S 2.90GHzprocessor), He et al. [21] acquired the running time, as shown in **Table 3**. To realize 1024-bit RSA security, they used a Tate pairing $\hat{e}: G_1 \times G_1 \to G_2$, where G_1 defined on a super singular curve E/F_p: $y^2 = x^3 + 1$ is an additive group of q order, the lengths of q and p are 160 bits and 512 bits, separately.

Table 4. Comparison of three CLE schemes

Scheme	Hwang [16]	Cheng [19]	New scheme
Encrypt	$n\cdot A_{G_1}+3\cdot S_{G_1}$ $+M_{G_2}+E_{G_2}+H$	$n\cdot A_{G_1}+3\cdot S_{G_1}$ $+M_{G_2}+E_{G_2}+H$	$B_P+2\cdot A_{G_1}$ $+5\cdot S_{G_1}+3\cdot H$
Decrypt	$4\cdot B_P+(n+2)\cdot A_{G_1}$ $+2\cdot M_{G_2}+H$	$4\cdot B_P+(n+2)\cdot A_{G_1}$ $+3\cdot M_{G_2}+H$	$B_P+S_{G_1}+2\cdot H$

Time $(n=60)$	$0.026n+28.585$ (30.145)	$0.026n+28.586$ (30.146)	23.905
Size of *Params* $(n=60)$	$(2n+4)\lvert G_1\rvert+\lvert G_2\rvert$ $(8000\ bytes)$	$(2n+4)\lvert G_1\rvert+\lvert G_2\rvert$ $(8000\ bytes)$	$2\lvert G_1\rvert$ $(128\ bytes)$
Size of *Msk* $(n=60)$	$(2n+4)\lvert Z_q^*\rvert$ (2480 bytes)	$(2n+4)\lvert Z_q^*\rvert$ (2480 bytes)	$\lvert Z_q^*\rvert$ $(20\ bytes)$
Security	No	Yes	Yes

For reasonableness, we suppose that the size of the user's identity information is 60 bits, i.e. $n=60$. It is a reasonable assumption that $\lvert\Delta_{ID}\rvert=\left\lfloor\frac{n}{2}\right\rfloor$. We use an intuitive way to evaluate the calculation cost. In [16], encrypting a plaintext demands n addition operations in G_1, 3 scale multiplication operations in G_1, 1 multiplication operation in G_2, 1 exponentiation operation in G_2 and 1 general hash operation. Decryption a ciphertext requires 4 bilinear pairing operations, $n+2$ addition operations in G_1, 2 multiplication operations in G_2 and 1 general hash operation. So the resulting running time is

$$4\times5.427+(2\times60+2)\times0.013+3\times2.165+3\times0.001+0.339+2\times0.007=30.145\,\text{ms}.$$

In [19], encrypting a plaintext demands n addition operations in G_1, 3 scale multiplication operations in G_1, 1 multiplication operation in G_2, 1 exponentiation operation in G_2 and 1 general hash operation. Decrypting a ciphertext requires 4 bilinear pairing operations, $n+2$ addition operations in G_1, 3 multiplication operations in G_2 and 1 general hash operation. So the resulting running time is

$$4\times5.427+(2\times60+2)\times0.013+3\times2.165+4\times0.001+0.339+2\times0.007=30.146\,\text{ms}.$$

In the new scheme, encrypting a plaintext demands 1 bilinear pairing operation, 2 addition operations in G_1, 5 scale multiplication operations in G_1 and 3 general hash operations. Decrypting a ciphertext requires 1 bilinear pairing operation, 1 scale multiplication operations in G_1 and 2 general hash operations. So the resulting running time is

$$2\times5.427+2\times0.013+6\times2.165+5\times0.007=23.905\,\text{ms}.$$

Table 4 and **Fig. 2** illustrate the computation costs of three different CLE schemes.

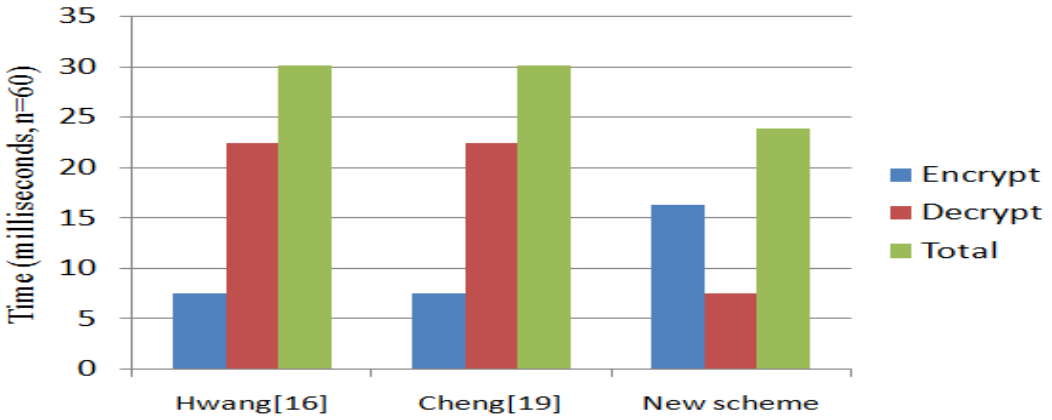

Fig. 2. Computation cost

Next, we evaluated the size of *params* and *msk*. In two schemes [16, 19], the *params* contain $2n+5$ points over a elliptic curve $E/F_p : y^2 = x^3 + 1$, thus the size is $\frac{(2\times 60+5)\times 512}{8} = 8000$ bytes. The *msk* contains $2n+4$ points in Z_q^*, thus the size is $\frac{(2\times 60+4)\times 160}{8} = 2480$ bytes. In the new scheme, the *params* contain only 2 points over a elliptic curve $E/F_p : y^2 = x^3 + 1$, thus the size is $\frac{2\times 512}{8} = 128$ bytes. The *msk* contains only one point in Z_q^*, thus the size is $\frac{160}{8} = 20$ bytes. **Table 4** and **Fig. 3** illustrate the storage costs of the three CLE schemes.

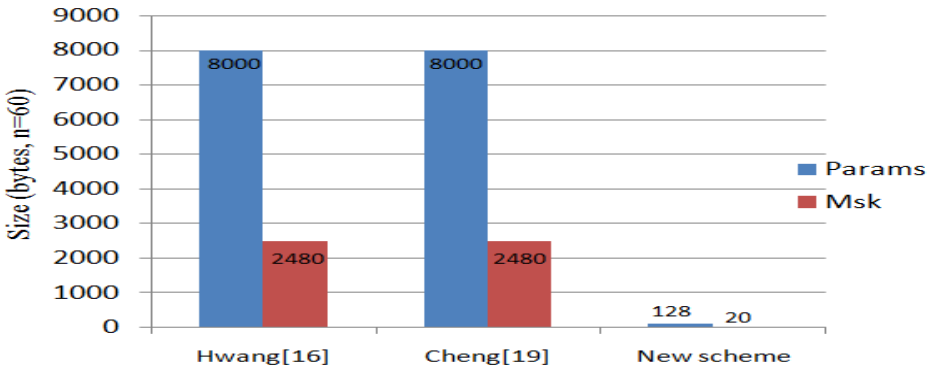

Fig. 3. Storage expenses

8. Conclusion

Certificateless cryptography is a significant technique to achieve data security and personal privacy protection. The security proofs of most known CLE schemes are done in ROM. It is well known, a cryptography scheme is not necessarily safe in real situations, even if its security proofs have been completed in ROM. There is only one CLE scheme [19], whose security proofs are completed in SM. But, the size of the storage space required by the system is linearly related to the size of the user's identity information. That adds to the storage burden of the key generation center. In this paper, we construct a fresh CLE scheme and complete the security proofs in SM. In the new scheme, it was constant that the size of the storage space required by the system, it was constant that the number of three kinds of operations (addition, scalar multiplication, and pairing). The computation cost and storage cost of the new scheme are lower than that of the previous ones.

Acknowledgment

The authors are grateful to the anonymous referees for their helpful comments and insightful suggestions. This research is supported by the National Natural Science Foundation of China under Grant No. 61962011, the Innovation Group Major Research Projects of Department of Education of Guizhou Province under Grant No.KY[2016]026. the Guizhou Provincial Science and Technology Foundation under Grant No.[2019]1434.

References

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

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

[3] Libert, B., and Quisquater, J., “On constructing certificateless cryptosystems from identity based encryption,” in *Proc. of International Workshop on Public Key Cryptography*, LNCS, vol.3958, pp.474-490, 2006. Article (CrossRef Link)

[4] Sun, Y., and Li, H., “short-ciphertext and BDH-based CCA2 secure certificateless encryption,” *Science China: Information Science*, vol.53, pp.2005-2015, 2010. Article (CrossRef Link)

[5] Baek, J., Safavi-Naini, R., and Susilo, W., “Certificateless public key encryption without pairing,” in *Proc. of International Conference on Information Security*, LNCS, vol.3650, pp.134-148, 2005. Article (CrossRef Link)

[6] Sun, Y., Zhang, F., and Baek, J., “Strongly secure certificateless public key encryption without pairing,” in *Proc. of International Conference on Cryptology and Network Security*, LNCS, vol.4856, pp.194-208, 2007. Article (CrossRef Link)

[7] Yan, X., Gong, P., Bai, Z., Wang, J., and Li, P., "New certificateless public key encryption scheme without pairing," *IET Information Security*, vol.7, iss.4, pp.271-276, 2013. Article (CrossRef Link)

[8] Guo, R., Wen, Q., Shi, H., Jin, Z., and Zhang, H, "An efficient and provably secure certificateless public key encryption scheme for telecare medicine information systems," *Journal of Medical Systems*, vol.37, no.5, pp.9965, 2013. Article (CrossRef Link)

[9] Deng, L., Zeng, J., Wang, X, "An improved certificateless encryption scheme for telecare medicine information systems," *Journal of Internet Technology*, vol.18, no.2, pp.223-227, 2017. Article (CrossRef Link)

[10] Zhou, Y., and Yang, B., "Leakage-resilient CCA2-secure certificateless public-key encryption scheme without bilinear pairing," *Information Processing Letters*, vol.130, pp.16-24, 2018. Article (CrossRef Link)

[11] SK Hafizul, I., Muhammad, K., and Ali M, Al., "Anonymous and provably secure certificateless multi receiver encryption without bilinear pairing," *Security and Communication Networks*, vol.8, pp.2214-2231, 2015. Article (CrossRef Link)

[12] He, D., Wang, H., Wang, L., Shen, J., and Yang, X., "Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices," *Soft Computing*, vol.21, no.22, pp.6801-6810, 2017. Article (CrossRef Link)

[13] Gao, R., Zeng, J., and Deng L., "Efficient certificateless anonymous multi-Receiver encryption scheme without bilinear parings," *Mathematical Problems in Engineering*, Article ID 1486437, 13 pages, 2018. Article (CrossRef Link)

[14] Huang, Q., and Wong, D.S., "Generic certificateless encryption in the standard model," in *Proc. of International Workshop on Security*, LNCS, vol.4752, pp.278-291, 2007. Article (CrossRef Link)

[15] Dent, A.W., Libert, B., and Paterson, K.G., "Certificateless encryption schemes strongly secure in the standard model," in *Proc. of International Workshop on Public Key Cryptography*, LNCS, vol.4939, pp.344-359, 2008. Article (CrossRef Link)

[16] Hwang, Y.H., Liu, J.K., and Chow, S.S, "Certificateless public key encryption secure against malicious KGC attacks in the standard model," *Journal of Universal Computer Science*, vol.14, no.3, pp.463-480, 2008. Article (CrossRef Link)

[17] Zhang, G., and Wang, X, "Certificateless encryption scheme secure in standard model," *Tsinghua Science & Technology*, vol.14, no.4, pp.452-459, 2009. Article (CrossRef Link)

[18] Shen, L., Zhang, F., Sun, Y., and Li, S., "Cryptanalysis of a certificateless encryption scheme in the standard model," in *Proc. of International Conference on Intelligent Networking and Collaborative Systems*, pp.329-333, 2012. Article (CrossRef Link)

[19] Cheng, L., Wen, Q., Jin, Z., and Zhang, H., "Cryptanalysis and improvement of a certificateless encryption scheme in the standard model," *Frontiers of Computer Science*, vol.8, no.1, pp.163-173, 2014. Article (CrossRef Link)

[20] Reza, S., Ron S., and Josef, Pieprzyk., "Lattice-based certificateless public-key encryption in the standard model," *International Journal of Information Security*, vol.13, pp.315-333, 2014. Article (CrossRef Link)

[21] He, D., Zeadally, S., Kumar, N., and Wu, W., "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE transactions on information forensics and security*, vol.11, no.9, pp.2052-2064, 2016. Article (CrossRef Link)

Lunzhi Deng received his B.S. from Guizhou Normal University, Guiyang, PR China, in 2002; M.S. from Guizhou Normal University, Guiyang, PR China, in 2008; and Ph.D. from Xiamen University, Xiamen, PR China, in 2012. He is now a professor in the School of Mathematical Sciences, Guizhou Normal University, Guiyang, PR China. His recent research interests include cryptography and information safety.
Email:denglunzhi@163.com

Tian Xia received his B.S. from Guizhou normal University, PR China, in 2009; he is now a graduate student at Guizhou Normal University in China. His recent research interests include digital signature and encryption protocols.

Xiuru He received her B.S. from Huaibei normal University, PR China, in 2018; She is now a graduate student at Guizhou Normal University in China. Her recent research interests include authentication protocol and information safety.