

Enhanced Certificate-Based Encryption Scheme without Bilinear Pairings

Yang Lu and Quanling Zhang

College of Computer and Information, Hohai University
Nanjing, Jiangsu 211100 - China
[e-mail: luyangnsd@163.com, zhangquanling99@163.com]
*Corresponding author: Yang Lu

*Received August 3, 2015; revised October 28, 2015; accepted December 23, 2015;
published February 29, 2016*

Abstract

Certificate-based cryptography is a useful public key cryptographic primitive that combines the merits of traditional public key cryptography and identity-based cryptography. It not only solves the key escrow problem inherent in identity-based cryptography, but also simplifies the cumbersome certificate management problem in traditional public key cryptography. In this paper, by giving a concrete attack, we first show that the certificate-based encryption scheme without bilinear pairings proposed by Yao *et al.* does not achieve either the chosen-ciphertext security or the weaker chosen-plaintext security. To overcome the security weakness in Yao *et al.*'s scheme, we propose an enhanced certificate-based encryption scheme that does not use the bilinear pairings. In the random oracle model, we formally prove it to be chosen-ciphertext secure under the computational Diffie-Hellman assumption. The experimental results show that the proposed scheme enjoys obvious advantage in the computation efficiency compared with the previous certificate-based encryption schemes. Without costly pairing operations, it is suitable to be employed on the computation-limited or power-constrained devices.

Keywords: Certificate-based cryptography, encryption, bilinear pairing, chosen-ciphertext security, random oracle model

This work is supported by the Nature Science Foundation of China under Grant No. 61272542 and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions. We would like to thank the anonymous referees for their helpful comments.

1. Introduction

Public key cryptography (PKC) is an important technique to realize network and information security. In PKC, each user has a pair of keys, namely a public key and a private key. The public key is usually published to the public while the corresponding private key is only known to its owner. However, in traditional PKC, each user's public key is generated randomly and does not contain any information associated with its owner. Therefore, it is infeasible to prove that a user is indeed the owner of a given public key. This problem can be solved by employing a trusted certification authority (CA) to generate public key certificates. A public key certificate is a digital signature issued by CA that binds a public key to the identity of its owner. By verifying a certificate, anyone can confirm whether a public key belongs to a user. This kind of certificate systems is referred to as public key infrastructure (PKI). However, the traditional PKI technology is faced with many practical challenges, especially the cumbersome certificate management problem. To eliminate the management of the certificates, Shamir [1] introduced the concept of identity-based cryptography (IBC) in 1984. In IBC, a user's public key is his unique identity information such as an e-mail address or a telephone number, and his private key is generated by a trusted third party called private key generator (PKG). Because the identity is a natural link to a user, the ability to use identities as public keys eliminates the need for public key certificates. However, IBC inevitably suffers from the key escrow problem as all users' private keys are known to the PKG.

In order to fill the gap between traditional PKC and IBC, Al-Riyami and Paterson [2] put forward the notion of certificateless public key cryptography (CL-PKC) in Asiacrypt 2003. In CL-PKC, a trusted third party called key generation center (KGC) is employed for generating a partial private key for each user. Each user generates a secret key and a public key, and then combines his secret key with the partial private key from the KGC to generate his full private key. Since KGC does not know any user's private key, CL-PKC overcomes the key escrow problem inherent in IBC. However, as partial private keys should be sent to users via secure channels, CL-PKC suffers from the key distribution problem. This drawback limits the application of CL-PKC in the public networks.

In Eurocrypt 2003, Gentry [3] proposed another new paradigm called certificate-based cryptography (CBC), which represents an interesting balance between IBC and traditional PKC. As in traditional PKC, each user in CBC first generates a private key and a public key, and then requests a certificate from a CA. The certificate in CBC acts not only as a public key certificate (as in traditional PKC) but also as a partial decryption/signing key, namely that each user should combine his private key with his certificate to generate his decryption/signing key. This additional functionality provides an effective implicit certificate mechanism so that a user needs both his private key and certificate to perform some cryptographic operations (such as decryption and signing), while the other communication parties need not obtain the fresh information on this user's certificate status. As a result, CBC eliminates the third-party query for the certificate status and simplifies the complicated certificate revocation problem in traditional PKC. As introduced by Gentry in [3], CBC can be used to construct efficient PKIs requiring fewer infrastructures than traditional ones. Furthermore, because the CA does not know the users' private keys and the certificates can be sent to the users publicly, CBC overcomes both the key escrow and key distribution problems. Following Gentry's pioneering work, numerous cryptographic schemes in the CBC setting (including many certificate-based encryption (CBE) schemes [4-11] and certificate-based signature (CBS) schemes [12-18])

have been proposed in the literature so far.

1.1 Motivation and contribution

The motivation of this paper is to develop a secure CBE scheme that does not depend on the costly bilinear pairings. As we know, compared with other common cryptographic operations such as scalar multiplications in the elliptic curve groups, the bilinear pairings may be the most expensive operations. Our experiment results show that the computation cost of the fastest Tate pairing is about 9 times as much as a scalar multiplication in the elliptic curve group under the 1024-bit RSA security level. As the computationally-heavy pairing operations will greatly aggravate the computation load of a device, they are extremely disliked by the computation-limited or power-constrained devices, such as smart phone, PDA. Therefore, as far as the efficiency, the cryptographic schemes without bilinear pairings would be more attractive. In 2013, Yao *et al.* [19] proposed a CBE scheme that does not use the bilinear pairings. They claimed that their scheme satisfies the chosen-ciphertext security in the random oracle model. Unfortunately, this is not true. Cryptanalysis shows that Yao *et al.*'s CBE scheme fails in achieving the chosen-ciphertext security, even the weaker chosen-plaintext security. The insecurity of Yao *et al.*'s scheme lies in the fact that an adversary can easily break the ciphertext indistinguishability of their scheme without making any oracle queries.

In this paper, we first give a concrete attack to show that Yao *et al.*'s CBE scheme [19] does not achieve their security claim. Based on the Schnorr signature scheme [20, 21] and the enhanced ElGamal public key encryption scheme proposed by Fujisaki and Okamoto [22], we newly develop a CBE scheme without depending on the bilinear pairings. Under the classic complexity assumption - computational Diffie-Hellman assumption, we formally prove that the proposed scheme satisfies the indistinguishable security under adaptive chosen-ciphertext attacks (*i.e.*, the chosen-ciphertext security) in the random oracle model [23, 24]. Compared with the previous pairing-based CBE schemes, the proposed scheme is more efficient in the computation efficiency. Due to avoiding the costly pairing operations, it is particularly suitable for the computation-limited or power-constrained devices.

1.2 Paper organization

The rest of this paper is organized as follows. In Section 2, we briefly review some related background definitions. In Section 3, we describe the attack against Yao *et al.*'s CBE scheme. The proposed CBE scheme is described and analyzed in Section 4. Finally, we draw our conclusion in Section 5.

2. Preliminaries

2.1 Elliptic curve group and complexity assumption

Let p be a prime number and F_p be a prime finite field. Let a and b be two elements such that $\Delta = 4a^3 + 27b^2 \neq 0$ in F_p . An elliptic curve over F_p (denoted by $E(F_p)$) defined by the parameters a and b is the set of all solutions $(x, y) \in F_p \times F_p$ to the equation $y^2 = x^3 + ax + b$, together with an extra point O at infinity. The set of points on $E(F_p)$ forms an abelian elliptic curve group

$$G = \{(x, y) \mid x, y \in F_p \wedge y^2 = x^3 + ax + b\} \cup \{O\}. \quad (1)$$

The point addition “+” in the elliptic curve group G is defined as follows: Let $P, Q \in G$, l_1 be the line connecting P and Q (l_1 be the tangent line to $E(F_p)$ if $P = Q$), and R be the third point of

intersection of the line l_1 with $E(F_p)$. Let l_2 be the line connecting R and O . Then $P + Q$ is the third point of intersection of the line l_2 with $E(F_p)$, namely $P + Q$ and R are x -axial symmetry points. The scalar multiplication in the group G can be computed as follows:

$$tP = P + P + \dots + P \text{ (} t \text{ times)}. \quad (2)$$

The security of the CBE scheme proposed in this paper relies on the computational Diffie-Hellman (CDH) assumption in the group G .

Definition 1. Given a tuple $(P, aP, bP) \in G^3$ for unknown $a, b \in Z_q^*$, the CDH problem in the group G is to compute $abP \in G$. The advantage of a probabilistic polynomial-time (PPT) algorithm A_{CDH} in solving the CDH problem is defined as

$$Adv(A_{CDH}) = Pr\{A_{CDH}(P, aP, bP) = abP \mid a, b \in Z_q^*\}. \quad (3)$$

The CDH assumption is that, for any PPT algorithm A_{CDH} , the advantage $Adv(A_{CDH})$ is negligible.

2.2 Formal model of certificate-based encryption

Usually, a CBE scheme is composed of five algorithms: (1) System setup algorithm **Setup**, which is performed by a CA to generate a master secret key msk and a set of public parameters $params$; (2) Key-pair generation algorithm **KeyPairGen**, which is performed by the user to generate a pair of private key and public key; (3) Certification algorithm **Certify**, which is performed by a CA to generate a certificate for each user in the system; (4) Encryption algorithm **Encrypt**, which is performed by the senders to encrypt the messages; (5) Decryption algorithm **Decrypt**, which is performed by the receivers to decrypt the received ciphertexts.

Fig. 1 gives the functional description of a CBE scheme.

-
- (1) **Setup**(k) $\rightarrow (msk, params)$
 Input: a security parameter $k \in Z^+$
 Output: a master key msk and a set of public parameters $params$
 - (2) **KeyPairGen**($params$) $\rightarrow (SK_U, PK_U)$
 Input: $params$
 Output: a private key SK_U and a public key PK_U for a user U
 - (3) **Certify**($params, msk, ID_U, PK_U$) $\rightarrow Cert_U$
 Input: $params, msk$, a user U 's identity ID_U and public key PK_U
 Output: a certificate $Cert_U$ for the user U
 - (4) **Encrypt**($params, M, ID_U, PK_U, Cert_U$) $\rightarrow C$
 Input: $params$, a message M , a receiver's identity ID_U , public key PK_U and certificate $Cert_U$
 Output: a ciphertext C
 - (5) **Decrypt**($params, C, ID_U, SK_U, Cert_U$) $\rightarrow M$
 Input: $params$, a ciphertext C , a receiver's identity ID_U , private key PK_U and certificate $Cert_U$
 Output: a message M or an error symbol \perp if C is an invalid ciphertext
-

Fig. 1. Functional description of a CBE scheme

Definition 2. A CBE scheme (**Setup**, **KeyPairGen**, **Certify**, **Encrypt**, **Decrypt**) is said to be correct if for any message M , $C = \text{Encrypt}(params, M, ID_U, PK_U, Cert_U)$, then $M =$

$\text{Decrypt}(params, C, ID_U, SK_U, Cert_U)$, where the public parameters $params$, the private/public key pair (SK_U, PK_U) and the certificate $Cert_U$ are respectively generated according to the specification of the algorithms **Setup**, **KeyPairGen** and **Certify**.

As introduced in [3, 6], the security model of CBE consists of two different types of adversaries, namely Type-I adversary (denoted by A_I) and Type-II adversary (denoted by A_{II}). The Type-I adversary A_I simulates an uncertified user who knows the target user's private key and can replace any user's public key, but cannot access the target user's certificate and the CA's master secret key, while the Type-II adversary A_{II} simulates a malicious CA in possession of the master secret key who can produce certificates for any users, but cannot access the target user's private key and replace any user's public key. It is clear that if a Type-II adversary A_{II} is allowed to replace public keys, by producing the certificates on the false public keys, then it can easily break any user's confidentiality.

In the security model of CBE, an adversary can make requests to some of the following five oracles adaptively. We assume that the challenger (or the simulator) keeps a history of "query-answer" when interacting with the adversaries. The oracles are described as follows:

(1) $O^{\text{CreateUser}}$: On input an identity ID_U , the challenger returns a public key PK_U . If the identity ID_U has not been created, then the challenger generates a set of private key SK_U , public key PK_U and certificate $Cert_U$ for the identity ID_U , and then returns PK_U as the output. In this case, ID_U is said to be created. Because the Type-II adversary A_{II} simulates a malicious CA who will generate a certificate for any user by itself, it is possible that the challenger is not aware of the secret(s) used by the adversary A_{II} to generate a certificate. Therefore, when creating a new user, the Type-II adversary A_{II} should submit the secret(s) to the challenger to generate a certificate for that user. For simplicity, we assume that other oracles defined below only respond to an identity which has been created.

(2) $O^{\text{ReplacePublicKey}}$: On input an identity ID_U and a public key PK'_U , the challenger replaces the current public key PK_U associated with the identity ID with PK'_U . This oracle is only queried by the Type-I adversary A_I , since the Type-II adversary A_{II} is disallowed to replace public keys.

(3) $O^{\text{PrivateKey}}$: On input an identity ID_U , the challenger responds with a private key SK_U associated with the identity ID_U . Here, the Type-I adversary A_I is disallowed to query this oracle on any identity for which the public key has been replaced. This restriction is imposed due to the fact that it is unreasonable to expect the challenger to be able to provide a private key of a user for which it does not know the private key.

(4) $O^{\text{Certificate}}$: On input an identity ID_U , the challenger responds with a certificate $Cert_U$ associated with the identity ID_U . This oracle is only queried by the Type-I adversary A_I as the Type-II adversary A_{II} can generate any user's certificate by itself. Here, the Type-I adversary A_I is disallowed to query this oracle on any identity for which the public key has been replaced. This restriction is imposed due to the fact that it is unreasonable to expect the challenger to be able to provide a certificate on a false public key.

(5) O^{Decrypt} : On input an identity ID_U and a ciphertext C , the challenger responds with the decryption of the ciphertext C .

The indistinguishable security under adaptive chosen-ciphertext attacks (*i.e.*, the IND-CCA2 security) for CBE schemes is defined by two adversarial games IND-CCA2-I and IND-CCA2-II (see Fig. 2), in which a challenger interacts with the Type-I adversary A_I and the Type-II adversary A_{II} respectively.

The game IND-CCA2-I is played between the challenger and the Type-I adversary A_I , in which *state* represents some state information, *Oracles-I* means that the adversary A_I can adaptively query the oracles $O^{\text{CreateUser}}$, $O^{\text{ReplacePublicKey}}$, $O^{\text{PrivateKey}}$, $O^{\text{Certificate}}$ and O^{Decrypt} with the

following constraints: (1) The target identity ID_{ch} cannot be submitted to the oracle $O^{Certificate}$; (2) The target identity and the challenge ciphertext (ID_{ch}, C_{ch}) cannot be submitted to the oracle $O^{Decrypt}$. The game IND-CCA2-II is played between the challenger and the Type-II adversary A_{II} , in which $state$ represents some state information, $Oracles-II$ means that the adversary A_{II} can adaptively query the oracles $O^{CreateUser}$, $O^{PrivateKey}$ and $O^{Decrypt}$ with the following constraints: (1) The target identity ID_{ch} cannot be submitted to the oracle $O^{PrivateKey}$; (2) The target identity and the challenge ciphertext (ID_{ch}, C_{ch}) cannot be submitted to the oracle $O^{Decrypt}$.

IND-CCA2-I:

1. **Setup**(k) $\rightarrow (params, msk)$
 2. $A_I^{Oracles-I}(k, params) \rightarrow (ID_{ch}, M_0, M_1, state)$
 3. **Encrypt**($params, M_\beta, ID_{ch}, PK_{ch}, Cert_{ch}$) $\rightarrow C_{ch}$
 4. $A_I^{Oracles-I}(C_{ch}, state) \rightarrow \beta'$
 5. Output β'
-

IND-CCA2-II:

1. **Setup**(k) $\rightarrow (params, msk)$
 2. $A_{II}^{Oracles-II}(k, params, msk) \rightarrow (ID_{ch}, M_0, M_1, state)$
 3. **Encrypt**($params, M_\beta, ID_{ch}, PK_{ch}, Cert_{ch}$) $\rightarrow C_{ch}$
 4. $A_{II}^{Oracles-II}(C_{ch}, state) \rightarrow \beta'$
 5. Output β'
-

Fig. 2. Adversarial games IND-CCA2-I and IND-CCA2-II

In both the games IND-CCA2-I and IND-CCA2-II, we say that an adversary wins the game if $\beta = \beta'$. The adversary's advantage in winning the game is defined to be

$$Adv(A_X) = 2|Pr\{\beta = \beta'\} - 1/2|, \quad (4)$$

where X is either I or II .

Definition 3. A CBE scheme satisfies the IND-CCA2 security if no PPT adversary has non-negligible advantage in the games IND-CCA2-I and IND-CCA2-II.

In the above definition, if the adversary is disallowed to make any queries to the oracle $O^{Decrypt}$, then we obtain the weaker chosen-plaintext security (*i.e.*, the IND-CPA security) for the CBE schemes.

3. Cryptanalysis of Yao et al.'s CBE Scheme

In this section, we show that the CBE scheme without pairings proposed by Yao et al. [19] can not achieve either the IND-CCA2 security or the IND-CPA security.

3.1 Review of Yao et al.'s CBE scheme

Yao et al.'s CBE scheme [19] consists of the following five algorithms:

- (1) **Setup**: Input a security parameter k . Generate two primes p and q such that $p = 2q + 1$. Pick a generator g of Z_p^* . Pick $x \in Z_q^*$ uniformly at random as master secret key $msk = x$, and compute master public key $mpk = g^x \bmod p$. Choose hash functions: $H_1: \{0,1\}^* \times Z_p^* \rightarrow Z_q^*$,

$H_2: Z_p^* \times Z_q^* \rightarrow Z_p^*$. The public parameters are $params = \{p, q, g, mpk, H_1, H_2\}$.

(2) **KeyPairGen**: Input the public parameters $params$. Pick $s \in Z_q^*$ at random as the user U 's private key SK_U and compute $PK_U = g^s \bmod p$ as the user U 's public key. Return the private key and public key pair $(SK_U, PK_U) = (s, g^s)$.

(3) **Certify**: Input $(params, msk, ID_U, PK_U)$. Pick $y \in Z_q^*$ at random, compute $cert_1 = g^y$, $cert_2 = y + xH_1(ID_U, PK_U)$ and $cert_3 = y + x(y + xH_1(ID_U, PK_U))$. Then it returns $Cert_U = (cert_1, cert_2, cert_3)$ as the certificate for the user U .

(4) **Encrypt**: Input $(params, M, ID_U, PK_U, Cert_U)$, check whether $g^{cert_3} \cdot (mpk)^{-cert_2} = cert_1$. Then randomly choose $r \in Z_q^*$, compute $C_1 = g^r$, $C_2 = M \cdot (PK_U)^r \cdot (mpk)^{rH_1(ID_U, PK_U)} \cdot (cert_1)^r$ and $C_3 = H_2(g^r, M)$. Output the ciphertext $C = (C_1, C_2, C_3)$.

(5) **Decrypt**: Input $(params, C, ID_U, SK_U, Cert_U)$, compute $M' = \frac{C_2}{C_1^{cert_2 + SK_U}}$. If $H_2(C_1, M') = C_3$, return M' . Otherwise return \perp indicating a decryption failure.

3.2 Cryptanalysis

In [19], Yao *et al.* claimed that their scheme achieves the IND-CCA2 security against both the Type-I adversary A_I and the Type-II adversary A_{II} . However, this is not true. The adversary A_I (or A_{II}) can easily break the ciphertext indistinguishability of Yao *et al.*'s scheme in the following way:

- Once the challenger starts the adversarial game, the adversary A_I (or A_{II}) enters the challenge phase directly without making any oracle queries. It submits an identity ID_{ch} , two length-equal messages M_0 and M_1 on which it wants to be challenged.
- The challenger randomly chooses a bit $\beta \in \{0, 1\}$, encrypts the message M_β to generate a challenge ciphertext $C^* = (C_1^*, C_2^*, C_3^*)$ and returns C^* to the adversary A_I (or A_{II}).
- On receiving the challenge ciphertext $C^* = (C_1^*, C_2^*, C_3^*)$, the adversary A_I (or A_{II}) checks whether $C_3^* = H_2(C_1^*, M_0)$ holds. It is easy to see that $\beta = 0$ if the equation holds or $\beta = 1$ otherwise. Thus, the adversary A_I (or A_{II}) can correctly determine the bit β . This means that the adversary A_I (or A_{II}) successfully breaks the ciphertext indistinguishability of Yao *et al.*'s scheme.

Since both the Type-I adversary A_I and the Type-II adversary A_{II} can carry out the above attack without making any oracle queries, Yao *et al.*'s scheme does not satisfy either the IND-CCA2 security or the weaker IND-CPA security against both A_I and A_{II} .

4. The Proposed CBE Scheme

In this section, we propose a new CBE scheme without bilinear pairing and prove it to achieve the IND-CCA2 security under the CDH assumption in the random oracle model.

4.1 Description of the scheme

Our CBE scheme is constructed by incorporating the Schnorr signature scheme [20, 21] into the enhanced ElGamal public key encryption scheme proposed by Fujisaki and Okamoto [22]. A formal description of the proposed scheme is as follows:

(1) **Setup**: The CA does the following: generate an additive cyclic group G over elliptic curve $E(F_p)$ as described in Section 2 and determine a generator P of the group G ; choose a random value $\alpha \in Z_q^*$ and compute $P_{pub} = \alpha P$; choose three cryptographic hash functions $H_1: \{0,1\}^* \times G \times G \rightarrow Z_q^*$, $H_2: \{0,1\}^{n+l} \times \{0,1\}^* \times G \times G \rightarrow Z_q^*$ and $H_3: G \rightarrow \{0,1\}^{n+l}$, where n and l denote the bit-length of a plaintext and a random bit string respectively; output the public parameters $params = \{E(F_p), G, q, P, P_{pub}, n, l, H_1, H_2, H_3\}$ and the master secret key $msk = \alpha$.

(2) **KeyPairGen**: A user U chooses a random value $x_U \in Z_q^*$ as his private key SK_U and computes his public key $PK_U = x_U P$.

(3) **Certify**: To generate a certificate for a user U with identity ID_U and public key PK_U , the CA chooses a random value $y_U \in Z_q^*$ and computes $Cert_U = (Cert_U^1, Cert_U^2) = (y_U P, y_U + \alpha h_U)$, where $h_U = H_1(ID_U, PK_U, Cert_U^1)$.

(4) **Encrypt**: To send a message $M \in \{0,1\}^n$ to a user U with identity ID_U , the sender does the following: choose a random bit $\delta \in \{0,1\}^l$ and compute $r = H_2(M, \delta, ID_U, PK_U)$; compute $X = rP$ and $Y = (M||\delta) \oplus H_3(rQ_U)$, where $Q_U = PK_U + Cert_U^1 + H_1(ID_U, PK_U, Cert_U^1)P_{pub}$; set $C = (X, Y)$ as the ciphertext.

(5) **Decrypt**: To decrypt a received ciphertext C , a user U does the following: parse the ciphertext C as (X, Y) and compute $M||\delta = Y \oplus H_3((SK_U + Cert_U^2)X)$; check whether $X = rP$ holds where $r = H_2(M, \delta, ID_U, PK_U)$; accept M if it holds or reject the decryption otherwise.

4.2 Correctness

Theorem 1. The proposed CBE scheme is correct.

Proof. This theorem can be easily proved by the following equations:

$$\begin{aligned}
 & Y \oplus H_3((SK_U + Cert_U^2)X) \\
 &= (M||\delta) \oplus H_3(rQ_U) \oplus H_3((x_U + y_U + \alpha H_1(ID_U, PK_U, Cert_U^1))rP) \\
 &= (M||\delta) \oplus H_3(rQ_U) \oplus H_3(r(x_U P + y_U P + H_1(ID_U, PK_U, Cert_U^1)\alpha P)) \\
 &= (M||\delta) \oplus H_3(rQ_U) \oplus H_3(r(PK_U + Cert_U^1 + H_1(ID_U, PK_U, Cert_U^1)P_{pub})) \\
 &= (M||\delta) \oplus H_3(rQ_U) \oplus H_3(rQ_U) \\
 &= M||\delta.
 \end{aligned}$$

4.3 Security

Theorem 2. In the random oracle model, the proposed CBE scheme achieves the IND-CCA2 security under the CDH assumption.

This theorem can be proved by combining the following two lemmas.

Lemma 1. Suppose that $H_1 \sim H_3$ are random oracles and A_I is a Type-I adversary against the IND-CCA2 security of our CBE scheme with advantage ε when running in time τ , making at most q_{cu} queries to the oracle $O^{CreateUser}$, q_{rp} queries to the oracle $O^{ReplacePublicKey}$, q_{pk} queries to the oracle $O^{PrivateKey}$, q_{cer} queries to the oracle $O^{Certificate}$, q_{dec} queries to the oracle $O^{Decrypt}$ and q_i queries to the random oracles H_i ($1 \leq i \leq 3$). Then there exists an algorithm A_{CDH} to solve the CDH problem in G with advantage

$$\varepsilon' \geq \frac{1}{q_3} \left(\frac{\varepsilon}{e(q_{rp} + q_{cer} + 1)} - \frac{q_{dec}}{2^l} - \frac{q_2}{2^l} \right) \quad (5)$$

and running time $\tau' \leq \tau + (q_1 + q_2 + q_3 + q_{rp} + q_{cer} + q_{pk})O(1) + q_{cu}(3\tau_m + O(1)) + q_{dec}(4\tau_m + O(1))$, where e is the Euler number and τ_m denotes the time for computing a scalar multiplication in G .

Proof. We show how to construct an algorithm A_{CDH} to solve the CDH problem from the Type-I adversary A_I . Assume that the algorithm A_{CDH} is given a random CDH problem instance (G, p, P, aP, bP) . Its goal is to compute abP by interacting with the adversary A_I as follows:

In the setup phase of the game, the algorithm A_{CDH} first sets $P_{pub} = aP$. It then starts the game IND-CCA2-I by supplying the adversary A_I with the public parameters $params = \{E(F_p), G, q, P, P_{pub}, n, l, H_1, H_2, H_3\}$, where $H_1 \sim H_3$ are random oracles controlled by the algorithm A_{CDH} . Note that the master secret key msk is the value a which is unknown to the algorithm A_{CDH} .

During the query-answer phase, the adversary A_I can adaptively make queries to the oracles $H_1, H_2, H_3, O^{CreateUser}, O^{ReplacePublicKey}, O^{PrivateKey}, O^{Certificate}$ and $O^{Decrypt}$. The algorithm A_{CDH} responds the adversary A_I 's various queries as follows:

H_1 queries: The algorithm A_{CDH} maintains a list H_1List of tuples $(ID_i, PK_i, Cert_i^1, h_1)$. On receiving such a query on $(ID_i, PK_i, Cert_i^1)$, the algorithm A_{CDH} checks whether the list H_1List contains a tuple $(ID_i, PK_i, Cert_i^1, h_1)$. If it does, the algorithm A_{CDH} outputs h_1 to the adversary A_I directly. Otherwise, it outputs a random value $h_1 \in Z_q^*$ to the adversary A_I and inserts a new tuple $(ID_i, PK_i, Cert_i^1, h_1)$ into the list H_1List .

H_2 queries: The algorithm A_{CDH} maintains a list H_2List of tuples $(M, \delta, ID_i, PK_i, h_2)$. On receiving such a query on (M, δ, ID_i, PK_i) , the algorithm A_{CDH} checks whether the list H_2List contains a tuple $(M, \delta, ID_i, PK_i, h_2)$. If it does, the algorithm A_{CDH} outputs h_2 to the adversary A_I directly. Otherwise, it outputs a random value $h_2 \in Z_q^*$ to the adversary A_I and inserts a new tuple $(M, \delta, ID_i, PK_i, h_2)$ into the list H_2List .

H_3 queries: The algorithm A_{CDH} maintains a list H_3List of tuples (R, h_3) . On receiving such a query on R , the algorithm A_{CDH} checks whether the list H_3List contains a tuple (R, h_3) . If it does, the algorithm A_{CDH} outputs h_3 to the adversary A_I directly. Otherwise, it outputs a random value $h_3 \in \{0, 1\}^{n+l}$ to the adversary A_I and inserts a new tuple (R, h_3) into the list H_3List .

$O^{CreateUser}$ queries: The algorithm A_{CDH} maintains a list $UserList$ of tuples $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$. On receiving such a query on ID_i , the algorithm A_{CDH} outputs PK_i to the adversary A_I directly if the list $UserList$ contains a tuple $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$. Otherwise, A_{CDH} picks a random coin $c_i \in \{0, 1\}$ so that $Pr\{c_i = 1\} = \gamma$ for some value γ that will be determined later and performs as follows:

- If $c_i = 1$, it first randomly chooses $x_i, y_i \in Z_q^*$, computes $PK_i = x_iP$ and sets $SK_i = x_i$. It then inserts a new tuple $(ID_i, PK_i, SK_i, y_i, \perp, c_i)$ into the list $UserList$ and outputs PK_i to the adversary A_I . Note that the certificate of the identity ID_i is $Cert_i = (Cert_i^1, Cert_i^2) = (y_iP, y_i + aH_1(ID_i, PK_i, Cert_i^1))$, where $Cert_i^2$ is unknown to the algorithm A_{CDH} .
- Else if $c_i = 0$, it randomly chooses $x_i, s_i, t_i \in Z_q^*$, sets $SK_i = x_i$, $PK_i = x_iP$ and $Cert_i = (Cert_i^1, Cert_i^2) = (t_iP - s_iP_{pub}, t_i)$. It then inserts $(ID_i, PK_i, Cert_i^1, s_i)$ and $(ID_i, PK_i, SK_i, \perp, Cert_i, c_i)$ into the lists H_1List and $UserList$ respectively and outputs PK_i .

$O^{ReplacePublicKey}$ queries: On receiving an identity ID_i and a false public key PK'_i , the algorithm A_{CDH} retrieves a tuple of the form $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$ from the list $UserList$. If $c_i = 1$, it aborts. Otherwise, it replaces the tuple with $(ID_i, PK'_i, \perp, y_i, \perp, c_i)$.

$O^{PrivateKey}$ queries: On receiving such a query on ID_i , the algorithm A_{CDH} retrieves a tuple of the form $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$ from the list $UserList$ and returns SK_i to the adversary A_I .

$O^{Certificate}$ queries: On receiving such a query on ID_i , the algorithm A_{CDH} retrieves a tuple of the form $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$ from the list $UserList$. If $c_i = 1$, A_{CDH} aborts. Otherwise, it returns $Cert_i$ to A_I .

$O^{Decrypt}$ queries: On receiving such a query on $(ID_i, C = (X, Y))$, the algorithm A_{CDH} performs as follows:

- If $c_i = 1$ or the public key associated with the identity ID_i has been replaced, it first runs the above simulation algorithm for the random oracle H_1 to get $h_1 = H_1(ID_i, PK_i, Cert_i^1)$, where $Cert_i^1 = y_i P$. It then checks if there exist a tuple $(M, \delta, ID_i, PK_i, h_2)$ in the list H_2List and a tuple $(h_2 Q_i, h_3)$ in the list H_3List such that $X = h_2 P$ and $Y = (M || \delta) \oplus h_3$, where $Q_i = PK_i + Cert_i^1 + h_1 P_{pub}$. If such two tuples exist, it returns M in the tuple $(M, \delta, ID_i, PK_i, h_2)$ to the adversary A_I as the decryption of the ciphertext C . Otherwise, it rejects this query. Note that a valid ciphertext is rejected with probability smaller than $q_{dec}/2^l$ across the whole game.
- Otherwise, it decrypts the ciphertext C in the normal way since it knows the private key SK_i and the certificate $Cert_i$ associated with the identity ID_i .

At the challenge phase, the adversary A_I outputs an identity ID_{ch} and two messages M_0, M_1 of equal length. The algorithm A_{CDH} retrieves a tuple of the form $(ID_{ch}, PK_{ch}, SK_{ch}, y_{ch}, Cert_{ch}, c_{ch})$ from the list $UserList$. If $c_{ch} = 0$, A_{CDH} aborts. Otherwise, it randomly chooses $\beta \in \{0, 1\}$ and $Y_{ch} \in \{0, 1\}^{n+l}$, sets $X_{ch} = bP$, and returns $C_{ch} = (X_{ch}, Y_{ch})$ as the challenge ciphertext to the adversary A_I . Observe that the decryption of the challenge ciphertext C_{ch} is $Y_{ch} \oplus H_3((SK_{ch} + Cert_{ch}^2)X_{ch}) = Y_{ch} \oplus H_3((x_{ch} + y_{ch} + aH_1(ID_{ch}, PK_{ch}, Cert_{ch}^1))bP)$ and $H_2(M_\beta, \delta^*, ID_{ch}, PK_{ch}) = b$, where $Cert_{ch}^1 = y_{ch} P$ and $\delta^* \in \{0, 1\}^l$.

At the guess phase, the adversary A_I outputs a bit β' which is ignored by the algorithm A_{CDH} . To produce a result, the algorithm A_{CDH} retrieves the secret values $SK_{ch} = x_{ch}$ and y_{ch} associated with the target identity ID_{ch} from the list $UserList$, randomly chooses a tuple $\langle R, h_3 \rangle$ from the list H_3List and computes

$$T = H_1(ID_{ch}, PK_{ch}, Cert_{ch}^1)^{-1}(R - x_{ch}bP - y_{ch}bP) \quad (6)$$

as the solution to the given CDH problem. It is easy to deduce that $T = abP$ if $R = (x_{ch} + y_{ch} + aH_1(ID_{ch}, PK_{ch}, Cert_{ch}^1))bP$.

This completes the simulation. We now estimate the advantage of the algorithm A_{CDH} in solving the given CDH problem. From the above construction, the simulation fails if any of the following events occurs:

- *Abort*: The algorithm A_{CDH} aborts during the simulation;
- *DecErr*: The algorithm A_{CDH} rejects a valid ciphertext submitted to the oracle $O^{Decrypt}$;
- *AskH₂^{*}*: The adversary A_I makes a query to the random oracle H_2 on $(M_\beta, \delta^*, ID_{ch}, PK_{ch})$;
- *AskH₃^{*}*: The adversary A_I makes a query to the random oracle H_3 on $(x_{ch} + y_{ch} + aH_1(ID_{ch}, PK_{ch}, Cert_{ch}^1))bP$.

Let $E = (DecErr \vee AskH_2^* \vee AskH_3^*) | \neg Abort$. It is clear that if the event E does not happen, then the adversary A_I does not gain any advantage greater than $1/2$ in the above simulation. Therefore, we get $Pr\{\beta' = \beta | \neg E\} \leq 1/2$. By splitting $Pr\{\beta' = \beta\}$, we have

$$\begin{aligned}
Pr\{\beta' = \beta\} &= Pr\{\beta' = \beta | \neg E\}Pr\{\neg E\} + Pr\{\beta' = \beta | E\}Pr\{E\} \\
&\leq Pr\{\neg E\}/2 + Pr\{E\} \\
&= 1/2 + Pr\{E\}/2.
\end{aligned}$$

Hence, we get $2|Pr\{\beta' = \beta\} - 1/2| \leq Pr\{E\}$. By the definition of the adversary A_I 's advantage in the game IND-CCA2-I, we have

$$\begin{aligned}
\varepsilon &= 2|Pr\{\beta' = \beta\} - 1/2| \\
&\leq Pr\{E\} \\
&\leq Pr\{(DecErr \vee AskH_2^* \vee AskH_3^*) | \neg Abort\} \\
&\leq (Pr\{DecErr\} + Pr\{AskH_2^*\} + Pr\{AskH_3^*\})/Pr\{\neg Abort\}.
\end{aligned}$$

We clearly have that $Pr\{\neg Abort\} = \gamma(1-\gamma)^q$, $Pr\{DecErr\} \leq q_{dec}/2^l$ and $Pr\{AskH_2^*\} \leq q_2/2^l$, where $q = q_{rp} + q_{cer}$ is the total number of A_I 's queries to the oracles $O^{ReplacePublicKey}$ and $O^{Certificate}$. The value of $\gamma(1-\gamma)^q$ is maximized at $\gamma_{max} = 1/(q+1)$. Thus, the probability that A_{CDH} does not abort is at least $1/(e(q+1))$. Therefore, we obtain

$$\begin{aligned}
Pr\{AskH_3^*\} &\geq Pr\{\neg Abort\}\varepsilon - Pr\{DecErr\} - Pr\{AskH_2^*\} \\
&\geq \varepsilon/(e(q+1)) - q_{dec}/2^l - q_2/2^l.
\end{aligned}$$

Since once the event $AskH_3^*$ happens, the algorithm A_{CDH} can solve the CDH problem by picking the correct tuple from the list H_3List . Therefore, we obtain the advantage of the algorithm A_{CDH} in solving the given CDH problem

$$\varepsilon' \geq \frac{1}{q_3} Pr\{AskH_3^*\} \geq \frac{1}{q_3} \left(\frac{\varepsilon}{e(q+1)} - \frac{q_{dec}}{2^l} - \frac{q_2}{2^l} \right). \quad (7)$$

From the simulation above, the running time of the algorithm A_{CDH} is bound by $\tau' \leq \tau + (q_1 + q_2 + q_3 + q_{rp} + q_{cer} + q_{pk})O(1) + q_{cu}(3\tau_m + O(1)) + q_{dec}(4\tau_m + O(1))$.

This completes the proof of Lemma 1. #

Lemma 2. Suppose that $H_1 \sim H_3$ are random oracles and A_{II} is a Type-II adversary against the IND-CCA2 security of our CBE scheme with advantage ε when running in time τ , making at most q_{cu} queries to the oracle $O^{CreateUser}$, q_{pk} queries to the oracle $O^{PrivateKey}$, q_{dec} queries to the oracle $O^{Decrypt}$ and q_i queries to the random oracles H_i ($1 \leq i \leq 3$). Then there exists an algorithm A_{CDH} to solve the CDH problem in G with advantage

$$\varepsilon' \geq \frac{1}{q_3} \left(\frac{\varepsilon}{e(q_{pk} + 1)} - \frac{q_{dec}}{2^l} - \frac{q_2}{2^l} \right) \quad (8)$$

and running time $\tau' \leq \tau + (q_1 + q_2 + q_3 + q_{pk})O(1) + q_{cu}(3\tau_m + O(1)) + q_{dec}(4\tau_m + O(1))$, where e is the Euler number and τ_m denotes the time for computing a scalar multiplication in G .

Proof. We show how to construct an algorithm A_{CDH} to solve the CDH problem from the Type-II adversary A_{II} . Assume that the algorithm A_{CDH} is given a random CDH problem instance (G, p, P, aP, bP) . Its goal is to compute abP by interacting with the adversary A_{II} as follows:

In the setup phase of the game, the algorithm A_{CDH} first randomly chooses a value $\alpha \in Z_q^*$. It then computes $P_{pub} = \alpha P$ and starts the game IND-CCA2-II by supplying the adversary A_{II} with the master key $msk = \alpha$ and the public parameters $params = \{E(F_p), G, q, P, P_{pub}, n, l, H_1,$

$H_2, H_3\}$, where $H_1 \sim H_3$ are random oracles controlled by the algorithm A_{CDH} .

During the query-answer phase, the adversary A_{II} can adaptively make queries to the oracles $H_1, H_2, H_3, O^{CreateUser}, O^{PrivateKey}$ and $O^{Decrypt}$. The algorithm A_{CDH} answers the adversary A_{II} 's queries to the oracles H_1, H_2, H_3 and $O^{Decrypt}$ in the same way as the proof of Lemma 1 and handles other oracle queries as follows:

$O^{CreateUser}$ queries: The algorithm A_{CDH} maintains a list *UserList* of tuples $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$. On receiving such a query on ID_i , it outputs PK_i to the adversary A_{II} directly if the list *UserList* contains a tuple $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$. Otherwise, after receiving a value $y_i \in Z_q^*$ from the adversary A_{II} , it picks a random coin $c_i \in \{0, 1\}$ so that $Pr\{c_i = 1\} = \gamma$ for some value γ and performs as follows:

- If $c_i = 1$, it first randomly chooses $h_i \in Z_q^*$, sets $PK_i = aP$ and $Cert_i = (Cert_i^1, Cert_i^2) = (y_iP, y_i + \alpha h_i)$. It then inserts new tuples $(ID_i, PK_i, Cert_i^1, h_i)$ and $(ID_i, PK_i, \perp, y_i, Cert_i, c_i)$ into the lists *H₁List* and *UserList* respectively and outputs PK_i to the adversary A_{II} . Note that the private key of the identity ID_i is $SK_i = a$ which is unknown to the algorithm A_{CDH} .
- Else if $c_i = 0$, it first randomly chooses $x_i, h_i \in Z_q^*$, sets $PK_i = x_iP, SK_i = x_i$ and $Cert_i = (Cert_i^1, Cert_i^2) = (y_iP, y_i + \alpha h_i)$. It then inserts $(ID_i, PK_i, Cert_i^1, h_i)$ and $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$ into the lists *H₁List* and *UserList* respectively and outputs PK_i to the adversary A_{II} .

$O^{PrivateKey}$ queries: On receiving such a query on ID_i , the algorithm A_{CDH} aborts if $c_i = 1$. Otherwise, it retrieves a tuple of the form $(ID_i, PK_i, SK_i, y_i, Cert_i, c_i)$ from the list *UserList* and returns SK_i to the adversary A_{II} .

At the challenge phase, the adversary A_{II} outputs two messages M_0 and M_1 of equal length and an identity ID_{ch} . The algorithm A_{CDH} retrieves a tuple of the form $(ID_{ch}, PK_{ch}, SK_{ch}, y_{ch}, Cert_{ch}, c_{ch})$ from the list *UserList*. If $c_{ch} = 0$, A_{CDH} aborts. Otherwise, it randomly chooses $\beta \in \{0, 1\}$ and $Y_{ch} \in \{0, 1\}^{n+l}$, sets $X_{ch} = bP$, and returns $C_{ch} = (X_{ch}, Y_{ch})$ as the challenge ciphertext to the adversary A_{II} . Observe that the decryption of the challenge ciphertext C_{ch} is $Y_{ch} \oplus H_3((SK_{ch} + Cert_{ch}^2)X_{ch}) = Y_{ch} \oplus H_3((a + y_{ch} + \alpha H_1(ID_{ch}, PK_{ch}, Cert_{ch}^1))bP)$ and $H_2(M_\beta, \delta^*, ID_{ch}, PK_{ch}) = b$, where $\delta^* \in \{0, 1\}^l$.

At the guess phase, the adversary A_{II} outputs a bit β' which is ignored by the algorithm A_{CDH} . To produce a result, the algorithm A_{CDH} retrieves the secret value y_{ch} associated with the identity ID_{ch} from the list *UserList*, randomly chooses a tuple (R, h_3) from the list *H₃List* and computes

$$T = R - y_{ch}bP - \alpha H_1(ID_{ch}, PK_{ch}, Cert_{ch}^1)bP \quad (9)$$

as the solution to the given CDH problem. It is easy to deduce that $T = abP$ if $R = (a + y_{ch} + \alpha H_1(ID_{ch}, PK_{ch}, Cert_{ch}^1))bP$.

As in the proof of Lemma 1, we can derive that the advantage of the algorithm A_{CDH} in solving the given CDH problem is bounded by

$$\varepsilon' \geq \frac{1}{q_3} \left(\frac{\varepsilon}{e(q+1)} - \frac{q_{dec}}{2^l} - \frac{q_2}{2^l} \right). \quad (10)$$

where $q = q_{pk}$ is the total number of A_{II} 's queries to the oracle $O^{PrivateKey}$.

From the simulation above, the running time of the algorithm A_{CDH} is bound by $\tau' \leq \tau + (q_1 + q_2 + q_3 + q_{pk})O(1) + q_{cu}(3\tau_m + O(1)) + q_{dec}(3\tau_m + O(1))$.

This completes the proof of Lemma 2. #

4.3 Efficiency comparison

Below, we make an efficiency comparison of our scheme and some previous CBE schemes [3, 6-9, 11] in terms of the encryption cost and the decryption cost.

We mainly consider six cryptographic operations: pairing, exponentiation in the bilinear target group G_2 , scalar multiplication in the bilinear group G_1 , scalar multiplication in the elliptic curve group G , map-to-point hash and general hash. Here (G_1, G_2) are the bilinear groups in the setting of bilinear pairing, *i.e.*, the bilinear pairing is $e: G_1 \times G_1 \rightarrow G_2$. For simplicity, we denote these operations by P, E, M_1, M, H_M and H respectively. Note that if G_1 is a multiplicative group, the scalar multiplication in G_1 is then called exponentiation correspondingly. The details of the compared CBE schemes are listed in Table 1. Note that we do not list all known CBE schemes in the literature but some secure and representative ones.

Table 1. Computation efficiency of the compared CBE schemes

Schemes	Encryption cost	Decryption cost
[3]	$2P+2E+1M_1+2H_M+3H$	$1P+1M_1+3H$
[6]	$8E+2M_1+1H_M+1H$	$2P+2E+1M_1+1H$
[7]	$2E+2M_1+4H$	$1P+1E+1M_1+3H$
[8]	$1P+1E+2M_1+1H_M+4H$	$2P+1E+2M_1+1H_M+4H$
[9]	$1E+5M_1+2H$	$4P+6M_1+2H$
[11]	$1E+4M_1+2H$	$2P+3M_1+H$
Ours	$3M+3H$	$2M+2H$

Table 2. Running time of the compared CBE schemes

Schemes	Encryption cost (ms)	Decryption cost (ms)
[3]	63.16	26.42
[6]	58.28	57.08
[7]	23.38	31.73
[8]	41.15	61.19
[9]	37.21	118.44
[11]	30.83	59.22
Ours	6.63	4.42

To give a more intuitive comparison, we simulate these CBE schemes using the standard cryptographic library MIRACAL [25] under the 1024-bit RSA security level. The experimental platform is a PIV 3-GHZ processor with 512-MB memory and a Windows XP operation system. For the pairing-based CBE schemes [3, 6-9, 11], to achieve the 1024-bit RSA security level, we use the fastest Tate pairing defined over the supersingular elliptic curve $E(F_p): y^2 = x^3 + x$ with embedding degree 2, where p is a 512-bit Solinas prime. For our scheme, to achieve the same security level, we use the security parameter secp160r1 recommended by the Standards for Efficient Cryptography Group (SECG) [26], where $p = 2^{160}$.

$-2^{31}-1$, $a = \text{FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 7FFFFFFC}$ and $b = \text{1C97BEFC 54BD7A8B 65ACF89F 81D4D4AD C565FA45}$. The simulation results are given in [Table 2](#).

As shown in [Table 2](#), the running time of the encryption algorithm of our scheme is 6.63ms which is about 10.5% of [\[3\]](#), 11.4% of [\[6\]](#), 28.4% of [\[7\]](#), 16.11% of [\[8\]](#), 17.8% of [\[9\]](#) and 21.5% of [\[11\]](#), while the running time of the decryption algorithm of our scheme is 4.42ms which is about 16.7% of [\[3\]](#), 7.7% of [\[6\]](#), 13.9% of [\[7\]](#), 7.2% of [\[8\]](#), 3.7% of [\[9\]](#) and 7.5% of [\[11\]](#). Therefore, it is more efficient than the previous pairing-based CBE schemes.

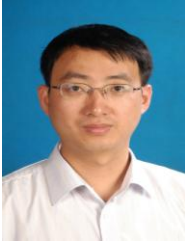
5. Conclusion

In this paper, we show that the CBE scheme without bilinear pairings proposed by Yao *et al.* [\[19\]](#) fails in achieving either the IND-CCA2 security or the weaker IND-CPA security. We propose an enhanced CBE scheme without relying on the bilinear pairings and formally prove that it satisfies the IND-CCA2 security under the CDH assumption in the random oracle model. Compared with the previous CBE schemes, our scheme enjoys obvious advantage in the computation efficiency. Due to avoiding the computationally-heavy pairing operations, it is suitable for the computation-limited or power-constrained devices.

References

- [1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of Advances in Cryptology - Crypto 1984*, pp. 47-53, August 19-22, 1984. [Article \(CrossRef Link\)](#).
- [2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. of Advances in Cryptology - Asiacrypt 2003*, pp. 452-473, November 30-December 4, 2003. [Article \(CrossRef Link\)](#).
- [3] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Proc. of Advances in Cryptology - Eurocrypt 2003*, pp. 272-293, May 4-8, 2003. [Article \(CrossRef Link\)](#).
- [4] C. Sur, C. D. Jung and K. H. Rhee, "Multi-receiver certificate-based encryption and application to public key broadcast encryption," in *Proc. of 2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security*, pp. 35-40, August 9-10, 2007. [Article \(CrossRef Link\)](#).
- [5] D. Galindo, P. Morillo and C. Ràfols, "Improved certificate-based encryption in the standard model," *Journal of Systems and Software*, vol. 81, no. 7, pp. 1218-1226, July, 2008. [Article \(CrossRef Link\)](#).
- [6] J. K. Liu and J. Zhou, "Efficient certificate-based encryption in the standard model," in *Proc. of 6th Int. Conf. on Security and Cryptography for Networks*, pp. 144-155, September 10-12, 2008. [Article \(CrossRef Link\)](#).
- [7] Y. Lu, J. Li and J. Xiao, "Constructing efficient certificate-based encryption with pairing," *Journal of Computers*, vol. 4, no. 1, pp. 19-26, January, 2009. [Article \(CrossRef Link\)](#).
- [8] Z. Shao, "Enhanced certificate-based encryption from pairings," *Computers and Electrical Engineering*, vol. 37, no. 2, pp. 136-146, March, 2011. [Article \(CrossRef Link\)](#).
- [9] W. Wu, Y. Mu, W. Susilo, X. Huang and L. Xu, "A provably secure construction of certificate-based encryption from certificateless encryption," *The Computer Journal*, vol. 55, no. 10, pp. 1157-1168, January, 2012. [Article \(CrossRef Link\)](#).
- [10] T. Hyla, W. Maćków and J. Pejaś, "Implicit and explicit certificates-based encryption scheme," in *Proc. of the 13th IFIP TC8 International Conference on Computer Information Systems and Industrial Management*, pp. 651-666, September 25-27, 2014. [Article \(CrossRef Link\)](#).
- [11] Y. Lu and J. Li, "Efficient construction of certificate-based encryption secure against public key replacement attacks in the standard model," *Journal of Information Science and Engineering*, vol. 30, no. 5, pp. 1553-1568, September, 2014. [Article \(CrossRef Link\)](#).
- [12] B. G. Kang, J. H. Park and S. G. Hahn, "A certificate-based signature scheme," in *Proc. of Topics in Cryptology - CT-RSA 2004*, pp. 99-111, February 23-27, 2004. [Article \(CrossRef Link\)](#).

- [13] M. H. Au, J. K. Liu, W. Susilo and T. H. Yuen, "Certificate based (linkable) ring signature," in *Proc. of 3rd Information Security Practice and Experience Conference*, pp.79-92, May 7-9, 2007. [Article \(CrossRef Link\)](#).
- [14] J. Li, X. Huang, Y. Mu, W. Susilo and Q. Wu, "Certificate-based signature: security model and efficient construction," in *Proc. of 4th European PKI Workshop Theory and Practice*, pp. 110-125, June 28-30, 2007. [Article \(CrossRef Link\)](#).
- [15] J. K. Liu, J. Baek, W. Susilo, and J. Zhou, "Certificate based signature schemes without pairings or random oracles," in *Proc. of 11th Information Security conference*, pp. 285-297, September 15-18, 2008. [Article \(CrossRef Link\)](#).
- [16] W. Wu, Y. Mu, W. Susilo, X. Huang, "Certificate-based signatures, revisited," *Journal of Universal Computer Science*, vol. 15, no. 8, pp. 1659-1684, April, 2009. [Article \(CrossRef Link\)](#).
- [17] J. Li, X. Huang, Y. Zhang and L. Xu, "An Efficient short certificate-based signature scheme," *Journal of Systems and Software*, vol. 85, no. 2, pp. 314-322, February, 2012. [Article \(CrossRef Link\)](#).
- [18] J. Li, Z. Wang and Y. Zhang, "Provably secure certificate-based signature scheme without pairings," *Information Science*, vol. 233, pp. 313-320, June, 2013. [Article \(CrossRef Link\)](#).
- [19] J. Yao, J. Li and Y. Zhang, "Certificate-based encryption scheme without pairing," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 6, pp. 1480-1491, June, 2013. [Article \(CrossRef Link\)](#).
- [20] C. P. Schnorr, "Efficient identifications and signatures for smart cards," in *Proc. of Advances in Cryptology - Crypto 1989*, pp. 239-252, August 20-24, 1989. [Article \(CrossRef Link\)](#).
- [21] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161-174, March, 1991. [Article \(CrossRef Link\)](#).
- [22] E. Fujisaki and T. Okamoto, "How to enhance the security of public-key encryption at minimum cost," in *Proc. of 2nd Int. Workshop on Theory and Practice in Public Key Cryptography*, pp. 53-68, March 1-3, 1999. [Article \(CrossRef Link\)](#).
- [23] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proc. of 1st ACM Conf. on Communications and Computer Security*, pp. 62-73, November 3-5, 1993. [Article \(CrossRef Link\)](#).
- [24] R. Canetti, O. Goldreich and S. Halevi, "The random oracle methodology, revisited," *Journal of ACM*, vol. 51, no. 4, pp. 209-218, July, 2004. [Article \(CrossRef Link\)](#).
- [25] MIRACL, Multiprecision integer and rational arithmetic cryptographic library, <http://certivox.org/display/EXT/MIRACL>
- [26] The Standards for Efficient Cryptography Group (SECG), SEC 2: Recommended elliptic curve domain parameters, Version 1.0, <http://www.secg.org/SEC2-Ver-1.0.pdf>.



Yang Lu received the Ph.D. degree from PLA University of Science and Technology in 2009. He has been working in HoHai University from 2003. Currently, he is an Assistant Professor in College of Computer and Information Engineering. His major research interests include information security and cryptography, network security and cloud security, etc. He has published more than 40 scientific papers in international conferences and journals.



Quanling Zhang has been studying in HoHai University from 2013. Currently, he is a postgraduate student in College of Computer and Information Engineering. His research interest is cryptography.