Certificate-Based Encryption Scheme without Pairing

Ji Yao, Jiguo Li and Yichen Zhang
College of Computer and Information Engineering, Hohai University
Nanjing 210098, China
[e-mail: ljg1688@163.com,lijiguo@hhu.edu.cn]
*Corresponding author: Jiguo Li

Received March 12, 2013; revised May 3, 2013; accepted May 31, 2013; published June 26, 2013

Abstract

Certificate-based cryptography is a new cryptographic primitive which eliminates the necessity of certificates in the traditional public key cryptography and simultaneously overcomes the inherent key escrow problem suffered in identity-based cryptography. However, to the best of our knowledge, all existed constructions of certificate-based encryption so far have to be based on the bilinear pairings. The pairing calculation is perceived to be expensive compared with normal operations such as modular exponentiations in finite fields. The costly pairing computation prevents it from wide application, especially for the computation limited wireless sensor networks. In order to improve efficiency, we propose a new certificate-based encryption scheme that does not depend on the pairing computation. Based on the decision Diffie-Hellman problem assumption, the scheme's security is proved to be against the chosen ciphertext attack in the random oracle. Performance comparisons show that our scheme outperforms the existing schemes.

Keywords: Public Key Cryptography, random oracle model, certificate-based encryption, without paring

This research was supported by the National Natural Science Foundation of China (60842002, 61272542, 61103183,61103184), the Fundamental Research Funds for the Central Universities(2009B21114, 2010B07114), China Postdoctoral Science Foundation Funded Project under Grant No. 20100471373, the Six Talent Peaks Program of Jiangsu Province of China (2009182) and Program for New Century Excellent Talents in Hohai University.

http://dx.doi.org/10.3837/tiis.2013.06.008

1. Introduction

The notion of certificate-based encryption (CBE) was first introduced by Gentry [1] in Eurocrpyt 2003. CBE combines traditional public key encryption (PKE) and identity-based encryption (IBE) while maintaining most of their characteristics. The certificate from a CBE scheme can be used not only proof of current certification but also acts as a partial decryption key. It gives us implicit certification that to eliminate third-party queries on certificate status. It also simplifies the public key revocation problem so that no infrastructures like CRL [2] and OCSP [3] are needed in CBE. Since the certificate authority does not know the private keys of users, it solves the key escrow problem. The key distribution problem is solved for the certificates need not be kept secret.

Based on Boneh and Franklin's [4] IBE scheme, Gentry [1] proposed first concrete CBE scheme. In EuroPKI 2004, Yum and Lee [5] proposed an equivalence theorem among IBE, certificate-less encryption (CLE) and CBE. They showed that CLE and CBE can be regarded as variants of IBE. Dodis and Katz [6] declared that a CBE scheme could be build by applying their generic techniques to an IBE and a PKE. Galindo et al. [7] pointed that the construction in [5] did not achieve the required security if CBE scheme, that is, a dishonest authority could break the security of the generic constructions. Lu et al. [8] solved this problem by providing two generic security-enhancing conversions based on the Fujisaki-Okamoto conversions [9,10] and proposed a method to achieve generic CBE constructions from PKE and IBE with CCA-secure in the random oracle model. Kang and Park [11] pointed out that the conversion proposed by Al-Riyami and Paterson [12] from CL-PKE to CBE is wrong. They said the conversions in [12] had a critical flaw in the security proof. Lu et al. [13] combined Sakai-Kasahara's IBE scheme [14,15] and traditional ElGamal-like cryptographic system [16] to constructed CBE scheme with pairing. Recently, Lu [17] proposed a new CBE scheme in the random oracle model. The security of scheme is under the hardness of the computational Diffie-Hellman problem and the gap bilinear Diffie-Hellman problem. In parallel to CBE, Kang et al. [18] proposed the security notion of certificate-based signature. Li et al. [19,20] formalized definition of the key replacement attack in certificate-based signature and refined the security model of certificate-based signature given in [18]. Furthermore they presented an efficient certificate-based signature scheme and proved it secure in the random oracle model. In order to improve performance of certificate-based signature, Li et al. [21,22] constructed an efficient short signature and a certificate-based signcryption with enhanced security features, respectively.

The above schemes were proved their securities in the random oracle model. However, Canetti et al. [23] declared that the schemes may not be secure when random oracles are instantiated with concrete hash functions. They suggested prove security of the schemes in the standard model. Based on the Waters scheme [24], Morillo and Ràfols [25] proposed the first concrete scheme in the standard model. Their model satisfies the minimal properties which are necessary to adapt the proof of [26] to obtain a fully secure CBE scheme. Galindo et al. [27] reviewed CBE schemes in the standard model and constructed a more efficient scheme. Liu and Zhou [28] constructed their scheme in the standard model which is motivated from Gentry's IBE scheme [29]. A generic construction of CBE scheme was proposed by Lu et al. [30] which is secure against adaptive chosen-ciphertext attacks.

Our contribution. Nevertheless, the above schemes all require pairing operations. According to MIRACL [31] achievement, a 512-bit Tate pairing takes 20 ms whereas a 1024-bit prime modular exponentiation takes 8.80 ms. The pairing computations are still considered as expensive comparing with normal operations. The costly pairing computation prevents it from wide application, especially for the computation limited wireless sensor networks. Recently, Li et al. presented a provably secure certificate-based signature scheme without pairing. However, no corresponding encryption scheme is proposed. In order to solve this problem, we construct a new certificate-based encryption scheme without pairing. Our scheme is proved secure against chosen ciphertext attack in the random oracle under the decision Diffie-Hellman problem.

Organization. In the rest of this paper, it is organized as follow. Section 2 gives some definitions and security models of CBE. The proposed scheme is presented in Section 3. In Section 4, we provide the security proof. We give performance comparison in Section 5. Finally, we conclude the paper in Section 6.

2. Preliminaries

In this section, we briefly review some definitions including hard problems, certificate-based encryption and secure model of CBE.

2.1 Decisional Diffie-Hellman Assumption (DDH)

Let p, q be primes such that $q \mid (p-1)$. Suppose g is an element selected from Z_p^* with order q. Let B be an attacker. B tries to solve the following problem: Given (g, g^a, g^b, T) for uniformly chosen $a, b \in_R Z_q^*$. B outputs 1 if $T = g^{ab}$ and 0 otherwise. We define B's advantage in solving the DDH problem is $Adv(B) = \Pr|B(g, g^a, g^b, T) = 1|$.

Definition 1. The decisional (t, ε) Diffie-Hellman assumption holds if no-t-time adversary has at least ε advantage in solving the above problem.

2.2 Certificate-Based Encryption

Recall the definitions of [1,25], the definitions for our CBE model is defined by five algorithms as follow:

- *Setup* is a probabilistic algorithm takes a security parameter k as input. It returns the certifier's master-key msk and the public parameters $params$ that including the description of message space $MSPC$ and ciphertext space $CSPC$.
- *SetKeyPair* is a probabilistic algorithm that takes $params$ as input. It returns user's private and public key pair (usk, upk).
- *Certify* is a probabilistic algorithm takes $\langle params, msk, \tau, id, upk \rangle$ as input. It returns $Cert$, which is sent to the user id through an open channel. Here τ is an index of the current time period.
- *Enc* is a probabilistic algorithm that takes $\langle params, \tau, id, upk, M \rangle$ as input. It returns a ciphertext $C \in CSPC$ for message M or $\perp$ indicating failure.
- *Dec* is a deterministic algorithm that takes $\langle params, \tau, Cert, usk, C \rangle$ as input. It returns either a message M or the special symbol $\perp$ indicating a decryption failure.

Naturally, it is required that for all M, $Dec(params, \tau, Cert, usk, Enc(params, \tau, id, upk, M)) = M$.

2.3 Secure Model of CBE

As defined in [17], we consider two types of adversaries for a CBE scheme, A_I for Type I and A_{II} for Type II. The adversary A_I essentially models an uncertified entity that has no access to the master key. It can gets any user's private key and gets certification with any identity except the challenge identity id^*. A_I also can request public key replace queries with values of its choice. The adversary A_{II} models the certifier in possession of the master key msk attacking an entity's public key. It can get any user's private key except the challenge user id^*. A_{II} also can requests public key replace queries with any user except the challenge identity id^*. The security model is defined with the help of two games as follow:

Game 1

- Setup: The challenger runs $Setup(1^k)$, generates master key msk and public parameters $params$, gives $params$ to A_I and keeps msk to itself.
- Phase 1: A_I 's queries and the challenger's responses as follow.

Public Key Queries: On input an identity id, the challenger responds with the public key upk for id.

Private Key Queries: On input an identity id, the challenger responds with the private key usk for id.

Public Key Repalcement: On input $\langle id, usk', upk' \rangle$, the challenger replaces the current public key upk with upk' for id.

Certificate Queries: On input $\langle \tau, id, upk \rangle$, the challenger responds with the certificate $Cert$. If the identity id has no associated certificate in the time period τ, then the challenger runs $Certify$ algorithm to generate a certificate.

Decryption Queries: On input $\langle \tau, id, C \rangle$, the challenger responds the decryption of C under the private key that is associated with the current public key.

- Challenge: Once A_I decides that Phase 1 is over, it outputs the challenge identity id^* and two equal-length plaintext messages M_0, M_1. Note that id^* has not been queried to certificate during the game. The challenger picks $\beta \in \{0,1\}$, runs the algorithm Enc. It takes $\langle params, \tau, id^*, upk_{id^*}, M_\beta \rangle$ as input, computes ciphertext

$$C^* = Enc(params, \tau, id^*, upk_{id^*}, M_\beta)$$

The challenge returns the ciphertext C^* to A_I.

- Phase 2: A_I makes queries as in Phase 1. But A_I cannot makes certificate query on $\langle \tau, id^*, upk_{id^*} \rangle$ and decryption query on the challenge ciphertext C^* for the combination (id^*, τ).
- Guess: Finally, A_I outputs a guess $\beta' \in \{0,1\}$.

We define A_I 's advantage in Game 1 is $Adv(A_I) = |2\Pr[\beta' = \beta] - 1| \geq \varepsilon$.

Definition 2. A CBE scheme is said to be IND-CCA2 secure if no probabilistic polynomial-time adversary A_I has non-negligible advantage ε in winning Game 1.

Game 2

- Setup: The challenger runs $Setup(1^k)$, generates master key msk and public parameters $params$, gives $(msk, params)$ to A_{II}.
- Phase 1: A_{II} 's queries and the challenger's responses as follow.

Public Key Queries: On input an identity id, the challenger responds with the public key upk for id.

Private Key Queries: On input an identity id, the challenger responds with the private key usk for id.

Public Key Repalcement: On input $\langle id, usk', upk' \rangle$, the challenger replaces the current public key upk with upk'.

Decryption Queries: On input $\langle \tau, id, C \rangle$, the challenger responds the decryption of C under the private key that is associated with the current public key.

- Challenge: Once A_{II} decides that Phase 1 is over, it outputs the challenge identity id^* and two equal-length plaintext messages M_0, M_1. Note that id^* has not been queried to private key and public key replacement. The challenger picks $\beta \in \{0,1\}$, and creates a target ciphertext

$$C^* = Enc(params, \tau, id^*, upk_{id^*}, M_\beta)$$

The challenge returns ciphertext C^* to A_{II}.

- Phase 2: A_{II} makes queries as in Phase 1. But A_{II} cannot makes a decryption query on the challenge ciphertext C^* for the combination (id^*, τ).
- Guess: Finally, A_{II} outputs a guess $\beta' \in \{0,1\}$.

We define A_{II} 's advantage in Game 2 is $Adv(A_{II}) = |2\Pr[\beta' = \beta] - 1| \geq \varepsilon$

Definition 3. A CBE scheme is said to be IND-CCA2 secure if no probabilistic polynomial-time adversary A_{II} has non-negligible advantage ε in winning Game 2.

3. An Efficient CBE Scheme

In this section, we construct a CBE scheme which is consisted of the following five algorithms:

- *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 $x \in_R Z_q^*$ uniformly at random as master secret key $msk = x$, and compute master public key $mpk = g^x \bmod p$. Choose hash functions: $H_1 : \{0,1\}^* \times Z_p^* \to Z_q^*$, $H_2 : Z_p^* \times Z_q^* \to Z_p^*$. The system parameters are $params = \{p, q, g, g^x, H_1, H_2\}$. The plaintext space $MSCP = Z_q^*$ and the ciphertext space $CSCP = Z_p^* \times Z_p^* \times Z_p^*$.
- *SetKeyPair*: Pick $s \in_R Z_q^*$ at random as user's secret key usk and compute $upk = g^s \bmod p$ as user's public key. Return the private/public key pair $(usk, upk) = (s, g^s)$ to user.
- *Certify*: Input $\langle params, msk, \tau, id, upk \rangle$. Pick $y \in_R Z_q^*$, compute $cert_1 = g^y$, $cert_2 = y + xh_1$, $cert_3 = y + x(y + xh_1)$, where $h_1 = H_1(id \| \tau, upk)$. Then it returns $Cert$ as the certificate for the identity id in the time period τ.

- *Enc*: Input $\langle params, \tau, id, upk \rangle$, check whether $g^{cert_3} \cdot (mpk)^{-cert_2} = cert_1$. Then choose a random string $r \in_R Z_q^*$, compute $C_1 = g^r$, $C_2 = M \cdot (upk)^r \cdot (mpk)^{h_1 r} \cdot (cert_1)^r$, $C_3 = H_2(g^r, M)$. Output the ciphertext $C = (C_1, C_2, C_3)$.
- *Dec*: Input $\langle params, Cert, usk, C \rangle$, compute $M' = \frac{C_2}{C_1^{cert_2 + usk}}$. If $H_2(C_1, M') = C_3$, return M'. Otherwise return $\perp$ indicating a decryption failure.

The correctness of the scheme is easy to check as we have

$$M' = \frac{C_2}{C_1^{cert_2+usk}} = \frac{M \cdot (upk)^r \cdot (mpk)^{h_1 r} \cdot (cert_1)^r}{(C_1)^{(y+xh_1)+s}} = \frac{M \cdot (g^s)^r \cdot (g^x)^{h_1 r} \cdot (g^y)^r}{(g^r)^{xh_1+y+s}} = M .$$

4. Security Analysis

Theorem 1. Suppose H_1, H_2 are random oracles and there exists a Type I IND-CCA2 adversary A_I against the CBE scheme with advantage ε, runs in time at most t, makes at most q_{pk} public key queries, q_{sk} private key queries, q_{pr} public key replace queries, q_c certificate queries, q_d decryption queries, q_1 times H_1 queries and q_2 times H_2 queries. Then there exists an algorithms B to solve the DDH problem running in time t' with advantage $\varepsilon' \geq \frac{\varepsilon+1}{2} \cdot \left(1 - \frac{1}{q_1}\right)^{q_c+q_d}$. The running time $t' \leq t + (q_1 + q_2 + q_{pr}) \cdot O(1) + (q_{pk} + q_{sk} + q_c) \cdot (t_{ex} + O(1)) + q_d \cdot (3t_{ex} + O(1))$, where t_{ex} denotes the time for computing exponentiation in Z_p^*.

Proof. Let A_I be a Type I adversary against the CBE scheme. We construct an algorithm B to solve the DDH problem. Given a random instance (g, g^a, g^b, T) of the DDH problem, we show how B plays as a challenger to interact with A_I, and solve the DDH problem with the ability of A_I.

Setup: B randomly chooses an index I with $1 \leq I \leq q_1$. B simulates the *Setup* algorithm, picks $x \in_R Z_q^*$ as msk and compute $mpk = g^x$. Supply A_I with the public parameters $\{p, q, g, g^x, H_1, H_2\}$, where H_1, H_2 are random oracles controlled by B. A_I may make queries to random oracles $H_i (i = 1, 2)$ at any time during its attack and B responds as follows:

H_1 *Queries*: B maintains a H_1 *list* of tuples $\langle (id, \tau, upk), e \rangle$, on receiving such a query on (id, τ, upk), if there is a tuple $\langle (id, \tau, upk), e \rangle$ on H_1 *list*, B returns e as answer. Otherwise, B chooses $e \in_R Z_q^*$, adds $\langle (id, \tau, upk), e \rangle$ to H_1 *list* and returns e as answer.

H_2 *Queries*: B maintains a H_2 *list* of tuples $\langle (A, M), w \rangle$, on receiving such a query on (A, M), if there is a tuple $\langle (A, M), w \rangle$ on H_2 *list*, B returns w. Otherwise, B chooses $w \in_R Z_q^*$, adds $\langle (A, M), w \rangle$ to H_2 *list* and returns w as answer.

- **Phase 1:** B maintains two lists of tuples $KeyList: \{id, usk, upk, \delta\}$ and $CertList: \{\tau, id, Cert\}$. The $KeyList$ is initiated empty, the $CertList$ is initiated with $\{\tau, id_I, (g^a, \perp, \perp)\}$. A_I launches Phase 1 of its attack by making a series of requests, and B responds as follows:

Public Key Queries : On receiving user's identity id_i , B searches $\{id_i, usk, upk, \delta\}$ on *KeyList* . If the tuple exists, B returns upk. Otherwise, B picks $s \in_R Z_q^*$ as usk and computes $upk = g^s$, adds $\{id_i, s, g^s, 0\}$ to *KeyList* and returns g^s .

Private Key Queries : On receiving user's identity id_i , B searches $\{id_i, usk, upk, \delta\}$ on *KeyList* . If the tuple exists, B returns usk . Otherwise, B picks $s \in_R Z_q^*$ as usk and computes $upk = g^s$, adds $\{id_i, s, g^s, 0\}$ to *KeyList* and returns s .

Public Key Replace : On receiving (id_i, usk', upk'), B searches id on *KeyList* and updates $\{id_i, usk', upk', 1\}$ on it.

Certificate Queries : On receiving (τ, id_i) , if $i = I$, B aborts. Then, if there is a tuple $CertList: \{\tau, id_i, Cert\}$ on the *CertList* , B returns *Cert* as answer. Otherwise, picks $y \in_R Z_q^*$, computes $h_1 = H_1(id_i \| \tau, upk)$, $Cert = (cert_1, cert_2, cert_3)$, $cert_1 = g^y$, $cert_2 = y + xh_1$, $cert_3 = y + x(y + xh_1)$. Adds $\{\tau, id_i, Cert\}$ to *CertList* . Returns *Cert* as answer.

Decryption Queries : On receiving (τ, id_i, C), if $i = I$, B aborts. Otherwise B computes $M' = \frac{C_2}{C_1^{cert_2 + usk}}$. If $H_2(C_1, M') = C_3$, returns M . Otherwise, outputs $\perp$.

- **Challenge:** A_I outputs id^*, τ and two messages M_0, M_1 on which it wishes to be challenged. If $id^* \neq id_I$, B aborts. Note that id^* had not been queried to certificate. B sets $C_1^* = g^b$, computer $C_2^* = M_\beta \cdot T \cdot g^{bxh^*} \cdot g^{bs^*}$, $C_3^* = H_2(g^r, M_\beta)$, outputs $C^* = (C_1^*, C_2^*, C_3^*)$.
- **Phase 2:** B continues to respond to A_I 's requests in the same way as it did in Phase 1. Note that A_I can not make a certificate query on (id^*, upk_{id^*}). No decryption query should be made on (id^*, C^*).
- **Guess:** Eventually, A_I outputs its guess β' . B outputs $T \neq g^{ab}$ if $\beta' \neq \beta$, else it outputs $T = g^{ab}$ as the solution to the DDH problem.

Analysis: From the simulation above, if $T = g^{ab}$, we have $C_2^* = M_\beta \cdot g^{ab} \cdot g^{xbh^*} \cdot g^{bs^*}$. Such that $C^* = (C_1^*, C_2^*, C_3^*)$ is a valid challenge ciphertext. And A_I outputs its guess $\beta' = \beta$ with advantage ε . Otherwise, A_I will not gain any advantage greater than $\frac{1}{2}$ to guess β'. Next, we estimate B's advantage in solving the DDH problem. Let $\neg Abort$ denotes the event that B does not abort during the simulation, *Solve* denotes the event that B solves the DDH problem when event $\neg Abort$ occurs. We obtain the probability $\Pr[\neg Abort] = \left(1 - \frac{1}{q_1}\right)^{q_c + q_d}$. By definition of ε , We have $\Pr[\beta' = \beta \mid T = g^{ab}] \geq \frac{1}{2} + \varepsilon$. And it is obvious that if event *Solve* does not happen during the simulation, B will not gain any advantage greater than $\frac{1}{2}$ to guess β , $\Pr[\beta' = \beta \mid T \neq g^{ab}] = \frac{1}{2}$. Then, we obtain

$$\Pr[Solve] = \Pr[\beta' = \beta \mid T = g^{ab}]\Pr[T = g^{ab}] + \Pr[\beta' = \beta \mid T \neq g^{ab}]\Pr[T \neq g^{ab}] \geq \left(\frac{1}{2} + \varepsilon\right) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{\varepsilon + 1}{2} .$$

Let E denotes the event that B solves the DDH problem. Then, we obtain
$\Pr[E] = \Pr[Solve] \cdot \Pr[\neg Abort] \geq \frac{\varepsilon+1}{2} \cdot \left(1 - \frac{1}{q_1}\right)^{q_c+q_d}$.

Therefore, we get B's advantage to solve the DDH problem $\varepsilon' \geq \frac{\varepsilon+1}{2} \cdot \left(1 - \frac{1}{q_1}\right)^{q_c+q_d}$. The running time $t' \leq t + (q_1 + q_2 + q_{pr}) \cdot O(1) + (q_{pk} + q_{sk} + q_c) \cdot (t_{ex} + O(1)) + q_d \cdot (3t_{ex} + O(1))$, where t_{ex} denotes the time for computing exponentiation in Z_p^*.

Theorem 2. Suppose H_1, H_2 are random oracles and there exists a Type II IND-CCA2 adversary A_{II} against the CBE scheme with advantage ε when running in time τ, making q_{pk} public key queries, q_{sk} private key queries, q_{pr} public key replace queries, q_d decryption queries q_1 times H_1 queries and q_2 times H_2 queries. Then there exists an algorithms B to solve the DDH problem in time t' with advantage $\varepsilon' \geq \frac{\varepsilon+1}{2} \cdot \left(1 - \frac{1}{q_1}\right)^{q_{sk}+q_d+q_{pr}}$. The running time $t' \leq t + (q_1 + q_2 + q_{pr}) \cdot O(1) + (q_{pk} + q_{sk} + q_c) \cdot (t_{ex} + O(1)) + q_d \cdot (3t_{ex} + O(1))$, where t_{ex} denotes the time for computing exponentiation in Z_p^*.

Proof. Let A_{II} be a Type II adversary against the CBE scheme. We construct an algorithm B to solve the DDH problem. Given a random instance (g, g^a, g^b, T) of the DDH problem, we show how B plays as a challenger to interact with A_{II}, and solve the DDH problem with the ability of A_{II}.

- **Setup:** B randomly chooses an index I with $1 \leq I \leq q_1$. B simulates the *Setup* algorithm, picks $x \in_R Z_q^*$ as msk and compute $mpk = g^x$. Supply A_{II} with msk and parameters $\{p, q, g, g^x, H_1, H_2\}$. A_{II} may make queries to random oracles $H_i(i = 1, 2)$ at any time during its attack and B responds as follows:

H_1 *Queries* : B maintains a H_1 *list* of tuples $\langle (id, \tau, upk), e \rangle$, on receiving such a query on (id, τ, upk), if there is a tuple $\langle (id, \tau, upk), e \rangle$ on H_1 *list* , B returns e as answer. Otherwise, B chooses $e \in_R Z_q^*$, adds $\langle (id, \tau, upk), e \rangle$ to H_1 *list* and returns e as answer.

H_2 *Queries* : B maintains a H_2 *list* of tuples $\langle (A, M), w \rangle$, on receiving such a query on (A, M), if there is a tuple $\langle (A, M), w \rangle$ on H_2 *list* , B returns w . Otherwise, B chooses $w \in_R Z_q^*$, adds $\langle (A, M), w \rangle$ to H_2 *list* and returns w as answer.

- **Phase 1:** B maintains a list of tuples $KeyList : \{id, usk, upk, \delta\}$, which is initiated with $\{id_I, \perp, g^a, \delta\}$. A_{II} launches **Phase 1** of its attack by making a series of requests, and B responds as follows:

Public Key Queries : On receiving (τ, id_i), if there is a tuple $\{id_i, usk, upk, \delta\}$ on $KeyList$, B returns upk. Otherwise, B chooses $s \in_R Z_q^*$ as usk and computes $upk = g^s$, add $\{id_i, s, g^s, 0\}$ to $KeyList$ and returns g^s as answer.

Private Key Queries : On receiving (τ, id_i) , if $i = I$, B aborts. Else if there is a tuple $\{id_i, usk, upk, \delta\}$ on $KeyList$, B returns usk . Otherwise, B chooses $s \in_R Z_q^*$ as usk and compute $upk = g^s$, add $\{id_i, s, g^s, 0\}$ to *KeyList* and returns s as answer.

Public Key Replace: On receiving (id_i, upk'), if $i = I$, B aborts. B searches id_i on *KeyList*, updates $\{id_i, usk, upk', 1\}$ on the list.

Decryption Queries: On receiving (τ, id_i, C), if $i = I$, B aborts. Otherwise B computes $M' = \frac{C_2}{C_1^{(xh_1+y)+usk}}$. If $H_2(C_1, M') = C_3$, return M. Otherwise, output $\perp$.

- **Challenge:** A_{II} outputs outputs id^*, τ and two messages M_0, M_1 on which it wishes to be challenged. Note that id^* had not been queried to private key and public key replacement. Otherwise, B sets $C_1^* = g^b$, computer $C_2^* = M_\beta \cdot T \cdot g^{bxh^*} \cdot g^{by^*}$, $C_3^* = H_2(g^r, M_\beta)$, outputs $C^* = (C_1^*, C_2^*, C_3^*)$.
- **Phase 2:** B continues to respond to A_{II} 's requests in the same way as it did in Phase 1. Note that A_{II} can not make a private key query or public key replace query on id^*. No decryption query should be made on (id^*, C^*).
- **Guess:** Eventually, A_{II} outputs its guess β'. B outputs $T \neq g^{ab}$ if $\beta' \neq \beta$, else it outputs $T = g^{ab}$ as the solution to the DDH problem.

Analysis: Using the same method in the proof of Theorem 1, the probability $\Pr[\neg Abort] = \left(1 - \frac{1}{q_1}\right)^{q_{sk}+q_d+q_{pr}}$. By definition of ε, We obtain $\Pr[\beta' = \beta \mid T = g^{ab}] \geq \frac{1}{2} + \varepsilon$. And it is obvious that if event *Solve* does not happen during the simulation, B will not gain any advantage greater than $\frac{1}{2}$ to guess β, $\Pr[\beta' = \beta \mid T \neq g^{ab}] = \frac{1}{2}$. Then, we obtain

$$\Pr[Solve] = \Pr[\beta' = \beta \mid T = g^{ab}]\Pr[T = g^{ab}] + \Pr[\beta' = \beta \mid T \neq g^{ab}]\Pr[T \neq g^{ab}] \geq \left(\frac{1}{2} + \varepsilon\right) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{\varepsilon + 1}{2}$$

. Let E denotes the event that B solves the DDH problem. Then, we obtain $\Pr[E] = \Pr[Solve] \cdot \Pr[\neg Abort] \geq \frac{\varepsilon + 1}{2} \cdot \left(1 - \frac{1}{q_1}\right)^{q_{sk}+q_d+q_{pr}}$.

Therefore, we get B's advantage to solve the DDH problem $\varepsilon' \geq \frac{\varepsilon + 1}{2} \cdot \left(1 - \frac{1}{q_1}\right)^{q_{sk}+q_d+q_{pr}}$. The running time $t' \leq t + (q_1 + q_2 + q_{pr}) \cdot O(1) + (q_{pk} + q_{sk} + q_c) \cdot (t_{ex} + O(1)) + q_d \cdot (3t_{ex} + O(1))$, where t_{ex} denotes the time for computing exponentiation in Z_p^*.

By combining Theorem 1 and Theorem 2, we can deduce that there is no probabilistic polynomial-time adversary can win either Game1 or Game 2 with non-negligible advantage in time t, i.e. our CBE scheme is IND-CCA2 secure in the random oracle model.

5. Performance Comparison

In this section, we will make a comparison of our scheme with the existing schemes. We consider four major operations: pairing (p), multiplication (m), exponentiation (e), hash (h).

Table 1. Performance comparison of the CBE schemes

Schemes	Encryption cost	Decryption cost
Ours	$3m+3e+1h$	$1m+1e+1h$
Scheme in [17]	$3m+1e+2h$	$1p+2m+1h$
Scheme in [13]	$2m+2e+4h$	$1p+1m+1e+3h$
Scheme in [22]	$5m+2e+4h$	$3p+3m$

From the table, our scheme requires two more exponentiation operation in the encryption algorithm. However, we do not use pairing operation in the decryption algorithm. Our scheme is still more efficient because the pairing operation is considered as the heaviest time-consuming operation according to MIRACL [31] achievement.

6. Conclusion

In this paper, we construct a new CBE scheme without pairings. We prove that our scheme is IND-CCA-secure in the random oracle. Security of scheme reduces to the hardness of the DDH problem. This makes our scheme possess strong applicability in applications where devices only have limited computational power (e.g. wireless sensor networks). In addition, currently most certificate-based signature schemes are secure in the random oracle model, but for which any implementation of the random oracle results in insecure schemes. To construct a CBE scheme without pairing in the standard model will be our future work.

References

[1] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Proc. of Eurocrypt 2003*, LNCS 2656, pp. 272-293, 2003. http://dx.doi.org/10.1007/3-540-39200-9_17

[2] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," *RFC 5280*, *IETF*, 2008. www.ietf.org/rfc/rfc5280.txt

[3] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet public key infrastructure online certificate status protocol – OCSP," *RFC 2560*, *IETF*, 1999. www.ietf.org/rfc/rfc2560.txt

[4] D. Boneh, M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. of Crypto 2001*, LNCS 2139, pp.213-229, 2001. http://dx.doi.org/10.1007/3-540-44647-8_13

[5] D. H. Yum, P. J. Lee, "Identity-based cryptography in public key management," in *Proc. of EuroPKI 2004*, LNCS 3093, pp.71-84, 2004. http://link.springer.com/chapter/10.1007%2F978-3-540-25980-0_6

[6] Y. Dodis, J. Katz, "Chosen-ciphertext security of multiple encryption," in *Proc. of TCC 2005*, LNCS 3378, pp.188-209, 2005. http://dx.doi.org/10.1007/978-3-540-30576-7_11

[7] D. Galindo, P. Morillo, C. Ràfols, "Breaking Yum and Lee generic construction of certificate-less and certificate-based encryption schemes," in *Proc. of EuroPKI 2006*, LNCS 4043, pp.81-91, 2006. dl.acm.org/citation.cfm?id=2107430

[8] Y. Lu, J. G. Li, J. M. Xiao, "Generic construction of certificate-based encryption," in *Proc. of 9th International Conference for Young Computer Scientists*, IEEE CS, pp.1589-1594, 2008. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4709210

[9] E. Fujisaki, T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Proc. of Crypto 1999*, LNCS 1666, pp. 537-554, 1999. http://link.springer.com/chapter/10.1007/3-540-48405-1_34#page-1

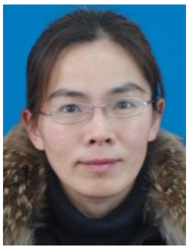
[10] E. Fujisaki, T. Okamoto, "How to enhance the security of public-key encryption at minimum cost,"

in *Proc. of PKC 1999*, LNCS 1560, pp. 53-68, 1999. dl.acm.org/citation.cfm?id=746447
[11] B.G. Kang, J.H. Park, “Is it possible to have CBE from CL-PKE,” in *Proc. of Cryptology ePrint Archive*, Report 2005/431. https://eprint.iacr.org/2005/431.pdf
[12] S. S. Al-Riyami, K. G. Paterson, “CBE from CL-PKE: A generic construction and efficient schemes,” in *Proc. of PKC 2005*, LNCS 3386, pp.398-415, 2005. http://dx.doi.org/10.1007/978-3-540-30580-4_27
[13] Y. Lu, J. G. Li, J. M. Xiao, “Constructing efficient certificate-based encryption with pairing,” *Journal of Computers*, vol. 4, no.1, pp.19-26, 2009. http://dx.doi.org/10.4304/jcp.4.1.19-26
[14] R. Sakai, M. Kasahara, “ID based cryptosystems with pairing on elliptic curve,” in *Proc. of Cryptology ePrint Archive*, Report 2003/054. https://eprint.iacr.org/2003/054.pdf
[15] L. Chen, Z. Cheng, “Security proof of Sakai-Kasahara's identity-based encryption scheme,” in *Proc. of Cryptography and Coding 2005*, LNCS 3796, pp. 442-459, 2005. http://link.springer.com/chapter/10.1007%2F978-3-540-30580-4_27
[16] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” in *Proc. of Crypto 1984*, LNCS 196, pp. 10-18, 1985. dl.acm.org/citation.cfm?id=19480
[17] Y. Lu, “An efficient and provably secure certificate-based encryption scheme,” in *Proc. of ICTMF 2011*, CCIS 164, pp. 54-61, 2011. http://dx.doi.org/10.1007/978-3-642-24999-0_8
[18] B. G. Kang, J. H. Park, S. G. Hahn, “A certificate-based signature scheme,” in *Proc. of CT-RSA*, LNCS 2964, pp. 99-111, 2004. http://dx.doi.org/10.1007/978-3-540-24660-2_8
[19] J. G. Li, X. Y. Huang, Y. Mu, W. Susilo, and Q. H. Wu, “Certificate-based signature: security model and efficient construction,” in *Proc. of EuroPKI’07*, LNCS 4582, pp.110-125, 2007. http://link.springer.com/chapter/10.1007/978-3-540-73408-6_8#page-1
[20] J. G. Li, X. Y. Huang, Y. Mu, W. Susilo, and Q. H. Wu, “Constructions of certificate-based signature secure against key replacement attacks,” *Journal of Computer Security*, vol.18, no.3, pp.421-449, 2010. dl.acm.org/citation.cfm?id=1835403
[21] J. G. Li, X. Y. Huang, Y. C. Zhang, L. Z. Xu, “An efficient short certificate-based signature scheme,” *Journal of Systems and Software*, vol.85, no.2, pp.314-322, 2012. http://dx.doi.org/10.1016/j.jss.2011.08.014
[22] J. G. Li, X. Y. Huang, M. X. Hong, Y. C. Zhang, “Certificate-based signcryption with enhanced security features,” *Computers and Mathematics with Applications*, vol.64, no.6, 1587-1601, 2012. http://dx.doi.org/10.1016/j.camwa.2012.01.006
[23] R. Canetti, O. Goldreich, S. Halevi, “The random oracle methodology, revisited,” *Journal of the ACM*, Vol. 51(4), pp. 557-594, 2004. http://dx.doi.org/10.1145/1008731.1008734
[24] B. Waters, “Efficient identity-based encryption without random oracles,” in *Proc. of Eurocrypt 2005*, LNCS 3494, pp.114-127, 2005. http://dx.doi.org/10.1007/11426639_7
[25] P. Morillo, C. Ràfols, “Certificate-based encryption without random oracles,” in *Proc. of Cryptology ePrint Archive*, Report 2006/12. https://eprint.iacr.org/2006/012.pdf
[26] D. Boneh, J. Katz, “Improved efficiency for CCA-secure cryptosystems built using identity-based encryption,” in *Proc. of CT-RSA 2005*, LNCS 3376, pp. 87-103, 2005. http://dx.doi.org/10.1007/978-3-540-30574-3_8
[27] D. Galindo, P. Morillo, C. Ràfols, “Improved certificate-based encryption in the standard model,” *Journal of Systems and Software*, vol. 81, pp. 1218-1226, 2008. http://dx.doi.org/10.1016/j.jss.2007.09.009
[28] J.K. Liu, J. Zhou, “Efficient certificate-based encryption in the standard model,” in *Proc. of SCN 2008*, LNCS 5229, pp. 144-155, 2008. dl.acm.org/citation.cfm?id=1432982
[29] C. Gentry, “Practical identity-based encryption without random oracles,” in *Proc. of Eurocrypt 2006*, LNCS 4004, pp. 445-464, 2006. http://dx.doi.org/10.1007/11761679_27
[30] Y. Lu, J. G. Li, “Generic construction of certificate-based encryption in the standard model,” in *Proc. of Electronic Commerce and Security 2009*, IEEE CS, Vol.1, pp. 25-29, 2009. dl.acm.org/citation.cfm?id=1606750.1607193
[31] MIRACL, “Multiprecision integer and rational arithmetic C/C++ library”. http://indigo.ie/ mscott/

[32] J. G. Li, Z. W. Wang, Y. C. Zhang, "Provably secure certificate-based signature scheme without pairings," *Information Sciences*, vol. 233, no. 6, pp. 313-320, 2013. http://dx.doi.org/10.1016/j.ins.2013.01.013

Yao Ji received his B.S. degree and M.S. degree in computer science and technology from Hohai University in 2008 and 2013, respectively. His research interests include cryptography, network security.

Jiguo Li received the B.S. degree in mathematics from Heilongjiang University, Harbin, China in 1996, M.S. degree in mathematics and PhD degree in computer science from Harbin Institute of Technology, Harbin, China in 2000 and 2003, respectively. He is a visiting scholar in the School of Computer Science and Software Engineering and the Centre for Computer and Information Security Research, University of Wollongong, Wollongong, Australia between 2006 and 2007. He is currently professor in the College of Computer and Information Engineering, Hohai University, Nanjing, China. He is currently holding the National Natural Science Foundation of China, the Fundamental Research Funds for the Central Universities, and the "Six Talent Peaks Program" of Jiangsu Province of China. His research interests include cryptography, network security, wireless security and trusted computing etc. He has published more than 80 referred research papers and two books. He has served on PC member of several international conferences and served as the reviewers of some international journal and conference.



Yichen Zhang received the B.S. degree in computer science from the Qiqihar University, Qiqihar, China in 1995. She is currently lecturer and PhD student in the College of Computer and Information Engineering, Hohai University, Nanjing, China. Her research interests include cryptography, network security. She has published more than 20 referred research papers at international conferences and journals.