Constructing Efficient Certificateless Public Key Encryption with Pairing

Yijuan Shi, Jianhua Li, and Jianjun Shi
(Corresponding author: Yijuan Shi)

Department of Electronic and Engineering, Jiao Tong University
Room 210, Building 2, Huashan Rd. 1954, Shanghai, 200030, China
(Email: cbzsyj130@sohu.com, {ijh888, jjshi}@sjtu.edu.cn)

(Received Jan. 4, 2006; revised and accepted Jan. 27 & Apr. 27, 2006)

Abstract

Certificateless public key cryptography was introduced to overcome the key escrow limitation of the identity-based cryptography. Recently, Yum1 and Lee have proposed a generic series construction model of certificateless public key encryption (CL-PKE). However, this model pays much attention on the generic construction and neglects the properties of the pairings. In this paper we propose a CL-PKE scheme which is based on the nice algebraic properties of the pairing. The scheme breaks through the old series model and works in an efficient parallel model. Our scheme is more efficient on computation and has more compact ciphertext than the existing schemes.

Keywords: Certificatelss public key encryption, parallel model, weil pairing

1 Introduction

Traditionally, a Public Key infrastructure (PKI) is used to provide an assurance to the user about the relationship between a public key and the identity of the holder of the corresponding private key by certificates. However, a PKI faces may challenges in the practice, especially the scalability of the infrastructure and the management of the certificates. To simplify the management of certificates, Shamir [11] proposed identity-based public key cryptography (ID-PKC) in which the public key of each party is derived directly from certain aspects of its identity, for example, an IP address belonging to a network host, or an e-mail address associated with a user. Private keys are generated for entities by a trusted third party called Key Generation Center (KGC). For a long while it was an open problem to obtain a secure and efficient identity based encryption (IBE) scheme. Until 2001, Boneh and Franklin [4] presented a provably secure identity-based encryption scheme (BF-IBE) using the bilinear pairings on elliptic curves. BF-IBE requires a special hash function which is probabilistic and generally inefficient. In 2003 Sakai and Kasahara [12] proposed another method of constructing identity-based keys, also using pairings, which has the potential to improve performance. This construction use general cryptographic hash functions rather than special ones. Later, Chen and Cheng [6] gave a provably secure identity-based scheme (SK-IBE) using this construction. The direct derivation of public keys in ID-PKC eliminates the need for certificates and some of the problems associated with them. However, the dependence on a KGC who can generate private keys inevitably introduces key escrow to the identity-based cryptography. Then in [1] Al-Riyami and Patersion introduced the notion of Certificateless Public Key Cryptography (CL-PKC). CL-PKC can overcome the key escrow limitation of ID-PKC without introducing certificates and the management overheads that this entails. It combines the advantages of the ID-PKC and the PKI.

In this paper, we concentrate on the certificateless public key encryption (CL-PKE) schemes. So far almost all the CL-PKE schemes [1, 2, 7, 8] are based on the BF-IBE scheme. Recently, Dae Hyun Yum and Pil Joong Lee [14] have proposed a generic series construction of CL-PKE which is built from generic primitives: identity-based encryption and public key encryption. The CL-PKE scheme in [2] is an instance of such model. However, this model pays much attention on the generic construction and neglects the nice properties of the bilinear pairings. In this paper, we propose an efficient CL-PKE scheme which is based on the nice algebraic properties of the bilinear pairing. The scheme works in a kind of parallel model and bases on the efficient identity-based encryption scheme SK-IBE [6] which requires only general hash functions. Hence our scheme does not requires special hash functions. Furthermore, our scheme is more efficient on computation and has more compact ciphertext than the existing schemes.

The paper is organized as follows: First we review the concepts of CL-PKE and two types of adversaries. In Section 3, we introduce some mathematic basis of bilinear maps. Then we present our new efficient CL-PKE scheme in Section 4 and analyze its security. In Section 5, we compare our scheme with the existing CL-PKE schemes on performance. Finally, Section 6 gives conclusions.

2 Certificateless Public Key Encryption

In this section, we review the definition and security model for CL-PKE from [1].

Definition 1. *[1] A CL-PKE scheme is specified by seven algorithms (Setup, Partial-Private-Key-Extract, Set-Secret-Value, Set-Private-Key, Set-Public-Key, Encrypt, Decrypt) such that:*

- ***Setup*** *is a probabilistic algorithm that takes security parameter κ as input and returns the system parameters params and the masterkey. The system parameters include a description of the message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$.*
- ***Partial-Private-Key-Extract*** *is a deterministic algorithm which takes params, masterkey and an identifier for entity A, $ID_A \in \{0,1\}^n$, as inputs. It returns a partial private key D_A .*
- ***Set-Secret-Value*** *is a probabilistic algorithm that takes as input params and outputs a secret value x_A.*
- ***Set-Private-Key*** *is a deterministic algorithm that takes params, D_A and x_A as inputs. The algorithm returns S_A, a (full) private key.*
- ***Set-Public-Key*** *is a deterministic algorithm that takes params and x_A as inputs and outputs a public key P_A.*
- ***Encrypt*** *is a probabilistic algorithm that takes params, $M \in \mathcal{M}$, x_A and ID_A as inputs and returns either a ciphertext $C \in \mathcal{C}$ or the null symbol $\perp$ indicating an encryption failure.*
- ***Decrypt*** *is a deterministic algorithm that takes as inputs params, $C \in \mathcal{C}$ and S_A. It returns a message $M \in \mathcal{M}$ or a message $\perp$ indicating a decryption failure.*

Algorithms **Set-Private-Key** and **Set-Public-Key** are normally run by an entity A for himself, after running **Set-Secret-Value**. Usually, A is the only entity in possession S_A and x_A. Algorithms **Setup** and **Partial-Private-Key-Extract** are usually run by a trusted third party, called Key Generation Center (KGC) [1].

Al-Riyami and Patersion presented the security model for CL-PKE in [1]. The security model distinguishes two types of adversaries:

Type I Adversary: Such an adversary $\mathcal{A}_I$ does not have access to the *masterkey*. However, $\mathcal{A}_I$ may request public keys and replace public keys with values of its choice, extract partial private and private keys and make decryption queries, all for identities of its choice.

Type II Adversary: Such an adversary $\mathcal{A}_{II}$ does have access to the *masterkey*, but may not replace public keys of entities. $\mathcal{A}_{II}$ can compute partial private keys for himself, given the *masterkey*. It can also request public keys, make private key extraction queries and decryption queries, both for identities of its choice. This adversary models security against an eavesdropping KGC.

3 Mathematic Basic

Before presenting the new CL-PKE scheme, we first review a few concepts related to bilinear maps. Let E/F_q be an elliptic curve and $m = \#E(F_q)$ be the group order of the curve. Let n be a prime such that $n \mid m$ and $n \nmid q$. Then the group of n-torsion points has the structure $E[n] \cong Z_n \oplus Z_n$ and is thus generated by two elements, say P_1 and P_2 ($< P_1 > \neq < P_2 >$) . We can denote the elements in the set of $E[n]$ using the form $aP_1 + bP_2$, $a, b \in Z_n^*$. Denote the group generated by P_1 by G_1 and the group generated by P_2 by G_2, i.e. $G_1 =< P_1 >$ and $G_2 =< P_2 >$. ψ is an isomorphism from G_2 to G_1 with $\psi(P_2) = P_1$. The Weil pairing is a function [10, 12]:

$$e_n : E[n] \times E[n] \rightarrow \mu_n.$$

e_n maps to the group μ_n of nth roots of unity, which is a cyclic group of order n as well. Denote this group by G_T. The following are some useful properties of the Weil Pairing.

- Identity: For all $P \in E[n]$, $e_n(P, P) = 1$.
- Alternation: For all $P, Q \in E[n]$, $e_n(P, Q) = e_n(Q, P)^{-1}$.
- Bilinearity: For all $P, Q, R \in E[n]$, $e_n(P + Q, R) = e_n(P, R)e_n(Q, R)$, and $e_n(P, Q + R) = e_n(P, Q)e_n(P, R)$.
- Non-degeneracy: For all $P \in G_1$ and $Q \in G_2$, $e_n(P, Q) \neq 1$.
- Computable: For all $P, Q \in E[n]$, $e_n(P, Q)$ is computable in polynomial time.

According to [13], we can either assume that the isomorphism ψ is computable in polynomial time or model the security proof with respect to a result whereby the adversary has access to an oracle which computes this isomorphism. In the following, we consider some problems.

co-BIDH Assumption: For $a, b, c \in_R Z_q^*, P_2 \in G_2^*, P_1 = \psi(P_2) \in G_1^*, e_n$, given (P_1, P_2, aP_2, bP_2), to compute $e_n(P_1, P_2)^{a^{-1}b}$ is hard.

k-BCAA1 Assumption: [6] For an integer k, and $x \in_R Z_n^*, P_2 \in G_2^*, P_1 = \psi(P_2) \in G_1^*, e_n$, given $(P_1, P_2, xP_2, h_0, (h_1, \frac{1}{h_1+x}P_2), ..., (h_k, \frac{1}{h_k+x}P_2))$ where $h_i \in_R Z_q^*$ and different from each other for $0 \le i \le k$, to compute $e_n(P_1, P_2)^{1/(x+h_0)}$ is hard.

k-BDHI Assumption: [5, 6] For an integer k, and $x \in_R Z_n^*, P_2 \in G_2^*, P_1 = \psi(P_2) \in G_1^*, e_n$, given ($P_1$, P_2, xP_2,x^2P_2,...x^kP_2), to compute $e_n(P_1,P_2)^{1/x}$ is hard.

The k-BDHI problem is well known [5, 6]. In [6] Chen and Cheng have proved the following relationship between the k-BCAA1 problem and the k-BDHI problem.

Theorem 1. *[6] If there exists a polynomial time algorithm to solve (k-1)-BDHI, then there exists a polynomial time algorithm for k-BCAA1. If there exists a polynomial time algorithm to solve (k-1)-BCAA1, then there exists a polynomial time algorithm for k-BDHI.*

From the Theorem 1, we know that the k-BCAA1 problem has a similar hardness with the k-BDHI problem. In the next section, we will present our new scheme which is based on the hardness of the k-BCAA1 problem.

4 A New CL-PKE Scheme

Inspired by the provable secure SK-IBE scheme [6, 12], we propose a new CL-PKE scheme. We describe our new scheme in a similar method of [4]. First, we give a basic CL-PKE scheme which is only IND-CPA secure. Then we will extend the basic scheme to the full scheme which is secure against an IND-CCA attack using a technique due to Fujisaki-Okamoto transformation [9].

4.1 Basic CL-PKE

Our basic scheme is consisted of the following algorithms.

Setup: Given a security parameter κ, the generator takes the following steps.

1) Generate a Weil pairing $e : E[q] \times E[q] \to G_T$ with $E[q] = G_1 \oplus G_2$ and an isomorphism ψ from G_2 to G_1. Pick a random generator $P_2 \in G_2^*$ and set $P_1 = \psi(P_2)$.

2) Pick a random $s \in Z_q^*$ and compute $P_{pub} = sP_1$.

3) Compute $g = e(P_1, P_2)$.

4) Pick cryptographic hash functions $H_1 : \{0,1\}^* \to Z_q^*$ and $H_2 : G_T \to \{0,1\}^n$.

The message space is $\mathcal{M} = \{0,1\}^n$. The ciphertext space is $\mathcal{C} = E[q] \times \{0,1\}^n$. The system parameters are *params* $=< q, G_1, G_2, G_T, e, n, P_1, P_2, g, P_{pub}, H_1, H_2 >$. The *masterkey* is s.

Partial-Private-Key-Extract: The algorithm takes as input an identifier $ID \in \{0,1\}^*$, *params* and the *masterkey* s and returns the partial private key $D_{ID} = \frac{1}{H_1(ID)+s}P_2$.

Set-Secret-Value: The algorithm takes as inputs *params* and identifier ID, selects a random $x_{ID} \in Z_q^*$ and outputs x_{ID} as the entity's secret value.

Set-Private-Key: The algorithm takes an inputs *params*, entity ID's partial private key D_{ID} and secret value x_{ID}. The output of the algorithm is the pair $S_{ID} =< D_{ID}, x_{ID} >$.

Set-Public-Key: The algorithm takes *params* and entity ID's secret value x_{ID} as inputs and constructs ID's public key as $P_{ID} = x_{ID}P_2$.

Encrypt: To encrypt $M \in \mathcal{M}$ for entity ID with the public key P_{ID}, perform the following steps:

1) Check that P_{ID} is in G_2^*, if not output $\perp$. This checks the validity of the public key.

2) Compute $Q_{ID} = H_1(ID)P_1 + P_{pub}$.

3) Choose random values r_1 and r_2 and compute the ciphertext:

$$C =< r_1Q_{ID} + r_2P_{ID}, M \oplus H_2(g^{(r_1+r_2)}) > .$$

Decrypt: Suppose $C =< U, V >$. To decrypt this ciphertext using the private key $S_{ID} =< D_{ID}, x_{ID} >$ compute:

$$M = V \oplus H_2(e(U, D_{ID} - \frac{1}{x_{ID}}P_1)).$$

According to the Weil Pairing's properties, we know $e(P_1, P_1) = 1$, $e(P_2, P_2) = 1$, and $e(P_2, -P_1) = e(P_1, P_2)$. Hence the consistency of the scheme can be verified by

$$\begin{aligned}
& e(U, D_{ID} - \frac{1}{x_{ID}}P_1) \\
= \;& e(r_1Q_{ID} + r_2P_{ID}, D_{ID} - \frac{1}{x_{ID}}P_1) \\
= \;& e(r_1(H_1(ID) + s)P_1 + r_2x_{ID}P_2, \\
& \frac{1}{H_1(ID) + s}P_2 - \frac{1}{x_{ID}}P_1) \\
= \;& e(r_1(H_1(ID) + s)P_1, \\
& \frac{1}{H_1(ID) + s}P_2)e(r_2x_{ID}P_2, -\frac{1}{x_{ID}}P_1) \\
= \;& e(P_1, P_2)^{r_1}e(P_1, P_2)^{r_2} \\
= \;& g^{(r_1+r_2)}.
\end{aligned}$$

4.2 Security of Basic CL-PKE

To study the security of the BasicCL-PKE scheme, we define the following two public key encryption schemes called BasicPub-I and BasicPub-II.

BasicPub-I: The scheme includes the following algorithms:

Key-generation: Given a security parameter κ, the generator takes the following steps.

1) Generate the parameters $< q, G_1, G_2, G_T, e, P_1, P_2, g >$ which are identical to the ones of the BasicCL-PKE.

2) Pick a random $s \in Z_q^*$ and compute $P_{pub} = sP_1$.Randomly choose different elements $h_i \in Z_q^*$ and compute $\frac{1}{h_i+s}P_2$ for for $0 \leq i < q_1$.

3) Pick a random $x \in Z_q^*$ and compute $P_{ID} = xP_2$.

4) Pick a hash function $H_2 : G_T \to \{0,1\}^n$.

The public parameters are $K_{pub-I} =< q, G_1, G_2, G_T, e, n, P_1, P_2, g, P_{pub}, x, P_{ID}, h_0, (h_1, \frac{1}{h_1+s}P_2), (h_2, \frac{1}{h_2+s}P_2), \cdots, (h_{q_1-1}, \frac{1}{h_{q_1-1}+s}P_2), H_2 >$ and the private key is $K_{pri-I} = \frac{1}{h_0+s}P_2$.

Encrypt: To encrypt $M \in \mathcal{M}$, perform the following steps:

1) Check that P_{ID} is in G_1^*, if not output $\perp$. This checks the validity of the public key.

2) Choose two random $r_1, r_2 \in Z_q^*$ and compute the ciphertext:

$$C =< r_1(h_0P_1 + P_{pub}) + r_2P_{ID}, M \oplus H_2(g^{(r_1+r_2)}).$$

Decrypt: Suppose $C =< U, V >$. To decrypt this ciphertext using the private key K_{pri-I} compute:

$$M = V \oplus H_2(e(U, K_{pri-I} - x^{-1}P_1)).$$

BasicPub-II: This scheme is similar to the BasicPub-I expect that s is publicly available, but x is kept secret.

Key-generation: Given a security parameter κ, the generator takes the following steps.

1) Generate the parameters $< q, G_1, G_2, G_T, e, P_1, P_2, g >$ which are identical to the ones of the BasicCL-PKE.

2) Pick a random $s \in Z_q^*$ and compute $P_{pub} = sP_1$. Randomly choose element $h_0 \in Z_q^*$.

3) Pick a random $x \in Z_q^*$ and compute $P_{ID} = xP_2$.

4) Pick a hash function $H_2 : G_T \to \{0,1\}^n$.

Hence the public parameters are $K_{pub-II} =< q, G_1, G_2, G_T, e, n, P_1, P_2, g, s, P_{pub}, P_{ID}, h_0, H_2 >$ and the private key is $K_{pri-II} = x$.

Encrypt: To encrypt $M \in \mathcal{M}$, perform the following steps:

1) Check that P_{ID} is in G_1^*, if not output $\perp$. This checks the validity of the public key.

2) Choose two random $r_1, r_2 \in Z_q^*$ and compute the ciphertext:

$$C =< r_1(h_0P_1 + P_{pub}) + r_2P_{ID}, M \oplus H_2(g^{(r_1+r_2)}).$$

Decrypt: Suppose $C =< U, V >$. To decrypt this ciphertext using the private key K_{pri-II} compute:

$$M = V \oplus H_2(e(U, \frac{1}{h_0 + s}P_2 - \frac{1}{K_{pri-II}}P_1)).$$

In the following, we prove that the BasicPub-I and BasicPub-II are IND-CPA secure.

Lemma 1. *The BasicPub-I scheme is secure against IND-CPA adversaries provided that H_2 is a random oracle and the k-BCAA1 assumption is sound.*

Proof. Algorithm $\mathcal{B}$ is given as input a random k-BCAA1 instance $< q, G_1, G_2, G_T, e, \psi, P_1, P_2, xP_2, h_0, (h_1, \frac{1}{h_1+x}P_2), \cdots, (h_{q_1-1}, \frac{1}{h_{q_1-1}+x}P_2) >$ where $x \in Z_q^*$ is a random element. Algorithm $\mathcal{B}$ finds $D = e(P_1, P_2)^{1/(x+h_0)}$ by interacting with $\mathcal{A}$ as follows:

Setup: Algorithm $\mathcal{B}$ first simulates algorithm Key-generation of BasicPub-I to create the public parameters as below.

1) Computes $P_{pub} = \psi(xP_2) \in G_1$.

2) Pick a random $r \in Z_q^*$ and set $P_{ID} = rP_2$.

3) Now $\mathcal{B}$ passes $\mathcal{A}$ the public parameters $K_{pub-I} =< q, G_1, G_2, G_T, e, \psi, P_1, P_2, P_{pub}, r, P_{ID}, h_0, (h_1, \frac{1}{h_1+x} P_2), \cdots, (h_{q_1-1}, \frac{1}{h_{q_1-1}+x}P_2) >$. The private key is $K_{pri-I} = \frac{1}{h_0+x}P_2$.

H_2-queries: At any time algorithm $\mathcal{A}$ can query the random oracle H_2. To response to these queries $\mathcal{B}$ maintains a list of tuples $< X_i, H_i >$. We refer to this list as the H_2^{list}. When $\mathcal{A}$ queries the oracle H_2 at a point X_i algorithm $\mathcal{B}$ responds as follows:

1) If the query X_i already appears on the H_2^{list} in a tuple $< X_i, H_i >$, then algorithm $\mathcal{B}$ responds with $H_2(X_i) = H_i$.

2) Otherwise, $\mathcal{B}$ chooses a random $H_i \in \{0,1\}^n$, return $H_2(X_i) = H_i$, and adds the tuple $< X_i, H_i >$ to the H_2^{list}.

Challenge: Algorithm $\mathcal{A}$ outputs two message M_0 and M_1 on which it wants to be challenged. $\mathcal{B}$ chooses a random string $R \in \{0,1\}^n$ and two random integers $r_1, r_2 \in Z_q^*$, and then defines the challenged ciphertext to be $C =< U, V >=< r_1P_1 + r_2P_{ID}, R >$. Observe that the decryption of C is $R \oplus H_2(r_1P_1 + r_2rP_2, \frac{1}{h_0+x}P_2 - r^{-1}P_1) = R \oplus H_2(D^{r_1} * e(P_1, P_2)^{r_2})$.

Guess: Algorithm $\mathcal{A}$ outputs it guess $b \in \{0,1\}$. At this point $\mathcal{B}$ pick a random tuple $< X_i, H_i >$ from the H_2^{list} and outputs $(X_i/e(P_1,P_2)^{r_2})^{-r_1}$ as the solution to the given instance of (q_1-1)-BCAA1 problem. □

Lemma 2. *The BasicPub-II scheme is secure against IND-CPA adversaries provided that H_2 is a random oracle and the co-BIDH assumption is sound.*

Proof. $\mathcal{B}$ is given as input a random co-BIDH problem instance $< P_1, P_2, aP_2, bP_2 >$. Let $D = e(P_1,P_2)^{a^{-1}b}$ be the solution to the co-BIDH problem. Algorithm $\mathcal{B}$ finds D by interacting with $\mathcal{A}$ as follows:

Setup: Algorithm $\mathcal{B}$ simulates algorithm Key-generation of the BasicPub-II to create the public $K_{pub-II} =< q, G_1, G_2, G_T, e, n, P_1, P_2, g, s, P_{pub}, P_{ID}, h_0, H_2 >$ by randomly selecting $s, h_0 \in Z_q^*$ and setting $P_{pub} = sP, P_{ID} = aP_2$. H_2 is a random oracle controlled by $\mathcal{B}$. The private key K_{pri-II} equals to a which $\mathcal{B}$ does not know. Then algorithm $\mathcal{B}$ passes the public key K_{pub-II} to $\mathcal{A}$ and responds queries as follows.

H_2-queries: To response to these queries $\mathcal{B}$ maintains a list of tuples $< X_i, H_i >$. We refer to this list as the H_2^{list}. When $\mathcal{A}$ queries the oracle H_2 at a point X_i algorithm $\mathcal{B}$ responds as follows:

1) If the query X_i already appears on the H_2^{list} in a tuple $< X_i, H_i >$, then algorithm $\mathcal{B}$ responds with $H_2(X_i) = H_i$.

2) Otherwise, $\mathcal{B}$ chooses a random $H_i \in \{0,1\}^n$, return $H_2(X_i) = H_i$, and adds the tuple $< X_i, H_i >$ to the H_2^{list}.

Challenge: Algorithm $\mathcal{A}$ outputs two message M_0 and M_1 on which it wants to be challenged. $\mathcal{B}$ chooses a random string $R \in \{0,1\}^n$ and a random integer $c \in Z_q^*$, and then defines the challenged ciphertext to be $C =< U, V >=< (h_0+s)cP_1 + bP_2, R >$. Observe that the decryption of C is $R \oplus H_2(e((h_0+s)cP_1 + bP_2, \frac{1}{h_0+s}P_2 - a^{-1}P_1) = R \oplus H_2(D * e(P_1,P_2)^c)$.

Guess: Algorithm $\mathcal{A}$ outputs it guess $b \in \{0,1\}$. At this point $\mathcal{B}$ pick a random tuple $< X_i, H_i >$ from the H_2^{list} and outputs $X_i/e(P_1,P_2)^c$ as the solution to the given instance of co-BIDH problem. □

According to the security of the above BasicPub-I and BasicPub-II schemes, we can prove the security of our new BasicCL-PKE scheme formally. For the limited space, we skip the detailed formal proof here and only analyze the security of our scheme heuristically for the two types of certificateless encryption adversaries.

Type I adversary $\mathcal{A}_I$: $\mathcal{A}_I$ does not know the *masterkey* s but he can replace public keys of entities with values of his choice. Suppose $\mathcal{A}_I$ selects $x \in Z_q^*$ randomly and replaces the public key of entity ID with $P'_{ID} = xQ_A$. If a sender wants to encrypt a message $M \in \mathcal{M}$ for entity ID, he computes the BasicCL-PKE ciphertext as:

$$C =< r_1Q_{ID} + r_2P'_{ID}, M \oplus H_2(g^{(r_1+r_2)}) > .$$

For the adversary $\mathcal{A}_I$ who knows x, the ciphertext C is the BasicPub-I encryption for the message M. Hence for the adversary $\mathcal{A}_I$ the IND-CPA security of the BasicCL-PKE scheme can be reduced to the IND-CPA security of the BasicPub-I scheme which is based on the hardness of the k-BCAA1 problem.

Type II adversary $\mathcal{A}_{II}$: $\mathcal{A}_{II}$ does have access to the *masterkey* s but he may not replace public keys of entities. With s, $\mathcal{A}_{II}$ can compute the partial private key D_{ID} for the entity ID. If a sender wants to encrypt a message $M \in \mathcal{M}$ for entity ID, he computes the CL-PKE ciphertext as:

$$C =< r_1Q_{ID} + r_2P_{ID}, M \oplus H_2(g^{(r_1+r_2)}) > .$$

For the $\mathcal{A}_{II}$ who knows the *masterkey* s, the ciphertext C is the BasicPub-II encryption for the message M. Hence for the adversary $\mathcal{A}_{II}$ the IND-CPA security of the BasicCL-PKE scheme can be reduced to the IND-CPA security of the BasicPub-II scheme which is based on the hardness of the co-BIDH problem.

4.3 FullCL-PKE

In this section, we use a technique due to Fujisaki and Okamoto [9] to convert the BasicCL-PKE scheme into an IND-CCA secure scheme. The Fujisaki-Okamoto transformation starts from an IND-CPA encryption scheme and builds an IND-CCA scheme in the random oracle model. Let $\mathcal{E}_{pk}(m,r)$ indicate the encryption of the indicated message m using the random bits r under the public key pk. The transformation is defined as:

$$\mathcal{E}_{pk}^{new}(m,r) = \mathcal{E}_{pk}((m \parallel r), H(m \parallel r)),$$

where r is a random string chosen from an appropriate domain and H denotes a hash function.

Lemma 3. *[9] Suppose that if $\mathcal{E}_{pk}$ is secure in the sense of IND-CPA, then $\mathcal{E}_{pk}^{new}$ obtained by the above transformation is secure in the sense of IND-CCA in the random oracle model.*

In the following, we apply the Fujisaki-Okamoto transformation to the BasicCL-PKE and then indicate that the resulting scheme called FullCL-PKE is IND-CCA secure. The FullCL-PKE is described as follows.

Setup: As in the BasicCL-PKE scheme. In addition, we select two hash functions $H_3 : \{0,1\}^* \to Z_q^*, H_4 : \{0,1\}^* \to Z_q^*$. Now $\mathcal{M} = \{0,1\}^{(n-k_0)}$ and $\mathcal{C} = E[q] \times \{0,1\}^n$.

Algorithms **Partial-Private-Key-Extract**, **Set-Secret-Value**, **Set-Private-Key** and **Set-Public-Key**

Table 1: Comparison of the CL-PKE schemes

Schemes	Encrypt	Decrypt	Pubkey Len
AP's Scheme I [1]	3p+1s+1e+4h*	1p+1s+3h	2
CC's Scheme I [7]	3p+1s+1e+4h*	1p+1s+3h	2
AP's Scheme II [2]	1p+2s+1e+5h*	1p+2s+4h	1
CC's Scheme II [8]	1p+2s+1e+4h*	1p+2s+3h	1
Our scheme	3s+1e+4h	1p+3s+3h	1

h*: Require a special hash function.

are identical to the ones of the BasicCL-PKE scheme.

Encrypt: To encrypt $M \in \{0,1\}^{(n-k_0)}$ for entity ID with the public key P_{ID}, perform the following steps:

1) Check that P_{ID} is in G_2^*, if not output $\perp$. This checks the validity of the public key.

2) Compute $Q_{ID} = H_1(ID)P_1 + P_{pub}$.

3) Choose a random $\sigma \in \{0,1\}^{k_0}$ and set $r_1 = H_3(M,\sigma), r_2 = H_4(M,\sigma)$.

4) Compute the ciphertext:

$$C = < r_1 Q_{ID} + r_2 P_{ID}, (M \parallel \sigma) \oplus H_2(g^{(r_1+r_2)}) > .$$

Decrypt: Suppose $C =< U, V >$. To decrypt this ciphertext using the private key $S_{ID} =< D_{ID}, x_{ID} >$ compute:

1) Compute $V \oplus H_2(e(U, D_{ID} - \frac{1}{x_{ID}} P_1)) = M \parallel \sigma$.

2) Parse $M \parallel \sigma$ and compute $r_1 = H_3(M,\sigma), r_2 = H_4(M,\sigma)$. Check that $U = r_1 Q_{ID} + r_2 P_{ID}$ where $Q_{ID} = H_1(ID)P_1 + P_{pub}$ can be precomputed. If not, reject the ciphertext.

3) Output M as the decryption of C.

The FullCL-PKE scheme is obtained by applying the above Fujisaki-Okamoto transformation to our IND-CPA secure BasicCL-PKE scheme. Then according to the lemma 3 we know that our FullCL-PKE is secure in the sense of IND-CCA in the random oracle model.

5 Performance Analysis

In this section, we will show that our proposed FullCL-PKE scheme has the best performance, comparing with other existing IND-CCA secure CL-PKE schemes [1, 2, 7, 8]. All the schemes have four major operations, i.e., Pairing (p), Scalar(s) and Exponentiation (e) and Hash (h). Pairing is the heaviest one even if many techniques have been applied on pairing operation to dramatically improve the performance[3].

In AP's Scheme I [1] and CC's Scheme I [7], the entity ID's public key has two elements of G_1. The validity test of the public key requires two pairing computations. Then their authors [2, 8] have improved their old schemes to Schemes II respectively. Public key has only one element of G_1 in AP's Scheme II and CC's Scheme II and the validity test of the public key is a simple group test $P_{ID} \in G_1$. AP's Scheme II and CC's Scheme II are more efficient than their old schemes for they require only 1 pairing operation while their old schemes require 3 pairing operations.

The advantage of our scheme is that it has better performance than the above existing schemes, particularly in encryption. First, the above existing schemes require a special hash function called MapToPoint [4] which maps an identifier to an element in G_1. The special hash function is generally inefficient and slower than the general hash function used in our scheme which maps an identifier to an element in Z_q^*. Second, no pairing operation is required in the Encrypt algorithm of our scheme. Even if our scheme requires 1 more scalar operation in Encrypt algorithm, it is still more efficient because pairing computation is much more time-consuming than scalar computation [3]. Finally, in any previous existing scheme, its ciphertext has three parts and the ciphertext space is $\mathcal{C} = G_1^* \times \{0,1\}^n \times \{0,1\}^n$. Compared with these schemes, our scheme has more compact ciphertext for it is consisted of only two parts and the ciphertext space is $\mathcal{C} = E[q] \times \{0,1\}^n$.

Without considering the pre-computation, the performance of the FullCL-PKE schemes are listed in Table 1, where we compare the schemes on the computation complexity and public key length (PK-Len). From Table 1, we can see that the computation complexity of our scheme compares favorably with previous known schemes.

6 Conclusions

In this paper, we present an efficient CL-PKE scheme. It has been analyzed to be IND-CCA secure in the random oracle model based on the hardness of the k-BCAA1 problem and the co-BIDH problem. Our scheme only requires generic hash functions rather than special ones. Compared with previous existing CL-PKE schemes, our scheme has absolute advantages in computation complexity and the length of the ciphertext.

References

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology ASIACRYPT'03*, LNCS 2894, pp. 452-473, Springer-verlag, 2003.

[2] S. S. Al-Riyami and K. G. Paterson, "CBE from CL-PKE" *A Generic Construciton and Efficient Schemes (PKC'05)*, LNCS 3386, pp. 398-415, 2005.

[3] P. S. L. M. Barreto, H. Y. Lynn, and M. Scott, " Efficient algorithms for pairing-based cryptosystems," in *Advances in Cryptology (Crypto'02)*, LNCS 2442, pp. 354-368, 2002.

[4] D. Boneh and M. Franklin, "Identity based encryption from the weil pairing," in *Advances in Cryptology (Crypto'01)*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

[5] D. Boneh, and X. Boyen, " Efficient selective-id secure identity-based encryption without random oracles," in *Proceedings of Advances in Cryptology (Eurocrypt'04)*, LNCS 3027, pp. 223-238, 2004.

[6] L. Q. Chen and Z. H. Cheng, *Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme*, Cryptology ePrint Archive, Report 2005/226.

[7] Z. H. Cheng, R. Comley, and L. Vasiu, "Remove key escrow from the identity-based encryption system," in *Foundations of Information Technology in the Era of Network and Mobile Computing*, pp. 37-50, France, Aug. 2004.

[8] Z. H. Cheng and R. Comley, *Efficient Certificateless Public Key Encryption*, Cryptology ePrint Archive, Report 2005/012.

[9] E. Fujisaki and T. Okamoto, " How to enhance the security of public-key encryption at minimum cost," *IEICE Transactions on Fundamentals*, E-83A, no. 1, pp. 24-32, 2000.

[10] M. Martijn, *Pairing-Based Cryptography*, Master thesis, Technische Universiteit Eindhoven, 2004.

[11] A. Shamir, "Identity based cryptosystems and signature schemes," in *Advances in Cryptology (Crypto'84)*, LNCS 196, pp. 47-53, Springer-Verlag, 1985.

[12] R. Sakai and M. Kasahara, *ID Based Cryptosystems with Pairing on Elliptic Curve*, Cryptology ePrint Archive, Report 2003/054.

[13] N. Smart and F. Vercauteren, *On Computable Isomorphisms in Efficient Pairing Based Systems*, Cryptology ePrint Archive, Report 2005/116.

[14] D. H. Yum and P. J. Lee, "Generic construction of certificateless encryption," in *Australasian Conference on Information Security and Privacy (ACISP'04)*, pp. 200-211, 2004.

Yijuan Shi received her M.S. in Communication and Information System from Electronic and Engineering Institute, Hefei, China. Currently, She is a Ph.D. candidate in Electronic Engineering, Shanghai Jiao Tong University (SJTU). Her current research interests include network management, network security and cryptography.

Jianhua Li received his M.S. and Ph.D. in Communication and Information System from the Shanghai Jiao Tong University, Shanghai, China. Currently, he is now a professor at the Department of Electronic and Engineering, Shanghai Jiao Tong University. His research interests include mobile communications, network management and information security.

Jianjun Shi received his M.S. and Ph.D. in Communication and Information System from the Shanghai Jiao Tong University. Currently, he is now an assistant professor at the Department of Electronic and Engineering, Shanghai Jiao Tong University. His research interests include mobile communications and network security.