

# An Authenticated Certificateless Public Key Encryption Scheme

Young-Ran Lee and Hyang-Sook Lee \*

Department of Mathematics, Ewha Womans University  
Seoul, 120-750, Korea, [sens1990@yahoo.co.kr](mailto:sens1990@yahoo.co.kr), [hsl@ewha.ac.kr](mailto:hsl@ewha.ac.kr)

**Abstract.** In 2003, Al-Riyami and Paterson [AP] proposed the certificateless public key cryptography(CL-PKC) which is intermediate between traditional certificated PKC and identity-based PKC. In this paper, we propose an authenticated certificateless public key encryption scheme. Our result improves their public key encryption scheme in efficiency and security. The security of the protocol is based on the hardness of two problems; the computational Diffie-Hellman problem(CDHP) and the bilinear Diffie-Hellman problem(BDHP). We also give a formal security model for both confidentiality and unforgeability, and then show that our scheme is provably secure in the random oracle model.

Keywords : certificateless public key encryption, confidentiality, unforgeability.

## 1 Introduction

The traditional public key cryptosystem has a well-established technology, public key infrastructure(PKI), but the issues of key management are somewhat complex. The problem of PKI technology are the certificate management including revocation, storage, distribution and the computational cost of certificate verification. In 1984, Shamir [Sh] proposed the concept of an identity-based cryptosystem to solve those key management problems. The idea of identity-based cryptosystems is to get rid of public key certificates by allowing the user's public key to be the binary sequence corresponding to an information identifying him in a non-ambiguous way.(E-mail address, IP address combined to a user name and social security number can be used.) Boneh-Franklin [BF1] [BF2] presented the first fully-functional and provably secure identity-based encryption scheme using bilinear maps over supersingular elliptic curves in 2001. The identity-based system needs a trusted private key generator(PKG) which generates the private keys of the entities using their public keys and a master secret key related to the global parameters

---

\* Supported by Korea Research Foundation Grant (KRF-2002-070-C00010)

for the system. Identity-based public key cryptosystems(ID-PKC) have an advantage in the aspect of the key management compared with the traditional public key system. However, ID-PKC has a significant shortcoming with respect to the PKG. The dependence on the PKG who can be in a privileged position by generating all user's private keys inevitably causes the key escrow problem to the ID-PKC. The issues of key escrow bring about several problems, for example, the invasion of a privacy, the dishonest PKG's masquerading as a regular user and so on.

Considering all these problems, Al-Riyami and Paterson [AP] introduced a new paradigm for public key cryptography, which is called a certificateless public key cryptography(CL-PKC). They proposed the certificateless public key encryption, signature, key exchange schemes and hierarchical CL-PKC. These schemes are all derived from pairings on elliptic curves. CL-PKC system does not require the use of certificate and does not have the key escrow feature of ID-PKC. Thus the CL-PKC is a model for the use of the public key cryptography that is intermediate between traditional PKI and ID-PKC.

In this paper we present an authenticated certificateless public key encryption scheme which improves the Al Riyami-Paterson's encryption scheme in efficiency and security. Our scheme also does not require certificates and does not suffer from the key escrow property that seems to be inherent in the identity-based PKC. To circumvent the escrow, the users in our model use the Diffie-Hellman key share from their ephemeral contributions that cannot be known by the PKG. In this scheme, the long-term keys are used for non-repudiation purpose and hence authentication. Consequently, our scheme keeps the confidentiality even from the PKG and gives the integrity. Furthermore, another advantage of our scheme would be damage control, in other words, disclosure of the master secret from the PKG would not compromise the confidentiality of the encrypted plaintext.

The security of our system is based on both the computational Diffie-Hellman (CDH) assumption and the bilinear Diffie-Hellman (BDH) assumption. Based on those assumptions, we first show that our scheme is EUF-CMA secure for integrity and then show that the scheme is IND-CCA secure for confidentiality.

The rest of our paper is organized as follows. In Section 2, we give underlying definitions and describe the security notions of our scheme. In Section 3, we present our public key encryption scheme which is not only provably secure against chosen ciphertext attack but also is existential unforgeable under adaptive chosen message attack in the random oracle

model, assuming that the CDH problem and the BDH problem are computationally hard. Furthermore, we analyze the security of our proposed scheme in Section 4. In Section 5, we compare the efficiency and the security of our proposed scheme with those of other known schemes. Finally, we give some conclusions in Section 6.

## 2 Preliminaries

### 2.1 Backgrounds

We first review the *admissible bilinear map*, which is the mathematical primitive that plays a central role in our public key encryption scheme.

*Bilinear map.* Let  $G_1$  denote an additive group of prime order  $q$  and  $G_2$  a multiplicative group of the same order. Let  $P$  be a generator of  $G_1$ . Assume that the discrete logarithm problem (DLP) is hard in both  $G_1$  and  $G_2$ .

A mapping  $\hat{e} : G_1 \times G_1 \rightarrow G_2$  satisfying the following properties is called an admissible bilinear map.

1. Bilinear:  $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$  for all  $P, Q \in G_1$  and  $a, b \in \mathbf{Z}_q^*$ .
2. Non-degenerate:  $\hat{e}$  does not send all pairs of points in  $G_1 \times G_1$  to the identity in  $G_2$ . (Hence, if  $P$  is a generator of  $G_1$  then  $\hat{e}(P, P)$  is a generator of  $G_2$ .)
3. Computable: There exists an efficient algorithm to compute  $\hat{e}(P, Q)$  for all  $P, Q \in G_1$ .

Typically the map  $\hat{e}$  will be derived from either Weil or Tate pairings on an elliptic curve over a finite field. The security of our scheme is based on the difficulty of computational Diffie-Hellman Problem (CDHP) and Bilinear Diffie-Hellman Problem (BDHP). Now we give formal descriptions of such hard problems.

*Computational Diffie-Hellman Problem.* Given  $P, aP, bP$  for some  $a, b \in \mathbf{Z}_q^*$ , compute  $abP$ .

*Computational Diffie-Hellman Assumption.* There exists no algorithm running in expected polynomial time which can solve the CDH problem with non-negligible probability.

*Bilinear Diffie-Hellman Problem.* Let  $G_1, G_2$  be two groups of prime order  $q$ . Let  $\hat{e} : G_1 \times G_1 \rightarrow G_2$  be an admissible bilinear map and let  $P$  be a generator of  $G_1$ . The BDH problem in  $\langle G_1, G_2, \hat{e} \rangle$  is as follows: Given  $\langle P, aP, bP, cP \rangle$  for some  $a, b, c \in \mathbf{Z}_q^*$ , compute  $W = \hat{e}(P, P)^{abc} \in G_2$ .

Algorithm  $\mathcal{A}$  has advantage  $\epsilon$  in solving BDHP in  $\langle G_1, G_2, \hat{e} \rangle$  if

$$\Pr[ \mathcal{A}(P, aP, bP, cP) = \hat{e}(P, P)^{abc} ] \geq \epsilon$$

where the probability is over the random choice of  $a, b, c$  in  $\mathbf{Z}_q^*$ , and the random bits of  $\mathcal{A}$ .

*Bilinear Diffie-Hellman Parameter Generator.* As in [BF1], [BF2], a randomized algorithm  $\mathcal{IG}$  is a BDH parameter generator if  $\mathcal{IG}$  takes a security parameter  $k > 0$ , runs in time polynomial in  $k$ , and outputs the description of an admissible pairing  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ .

*Bilinear Diffie-Hellman Assumption.* We say a BDH parameter generator  $\mathcal{IG}$ , satisfies the BDH assumption if the following is negligible in  $k$  for all probabilistic polynomial time algorithm  $\mathcal{A}$  :

$$\Pr [ (\mathbb{G}_1, \mathbb{G}_2, \hat{e}) \leftarrow \mathcal{IG}(1^k) ; P \leftarrow \mathbb{G}_1 ; a, b, c \leftarrow \mathbf{Z}_q^* : \mathcal{A}(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, aP, bP, cP) = \hat{e}(P, P)^{abc} ]$$

For the remainder of the paper we make use of some fixed BDH parameter generator  $\mathcal{IG}$  that satisfies the BDH assumption, and use the symbols  $\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q$  to represent the constituents of its output.

## 2.2 Security Notions

In the next section, we prove confidentiality and unforgeability of our scheme in the random oracle model based on the BDH assumption, CDH assumption and the Fujisaki-Okamoto transformation [FO]. We first give the formal definitions of confidentiality and unforgeability for our purpose.

### 2.2.1 Confidentiality

We say that a scheme is IND-CCA secure if no polynomially bounded adversary has a non-negligible advantage against the challenger in the following game.

**Setup** The challenger takes a security parameter  $k$  and runs the Setup algorithm to obtain parameters and a master key  $s$ . It gives the adversary parameters with the value  $s$  such that  $P_{pub} = sP$ .

Although our scheme is no longer ID based, a third party (PKG) issuing the long-term private keys of communicating parties exists in the system. To avoid the misbehavior of the PKG, our security model is strengthened more than the security models of other encryption schemes to handle adversaries. In short, we assume that the adversary can access to the master-key.

**Phase 1** The adversary issues queries  $q_1, q_2, \dots, q_m$  where  $q_i$  is one of ; *Extraction query of the form  $ID_I$* . On receiving such a query, the challenger runs  $\text{Extract}(ID_I)$  and responds with  $S_I = x_I d_I$ , where  $d_I$  is a long-term private key generated by the PKG.

*Encryption query of the form  $(ID_I, ID_J, M)$ .* On receiving such a query, the challenger runs  $\text{Extract}(ID_I) = S_I$  followed by  $\text{Encrypt}(S_I, ID_J, M)$ . The response is the resulting ciphertext.

**Decryption query of the form  $(ID_I, ID_J, C)$**  On receiving such a query the challenger runs  $\text{Extract}(ID_J)$  followed by  $\text{Decrypt}(ID_I, S_J, C)$ . The response is the plaintext  $M$ .

These queries may be asked adaptively, that is, each query  $q_i$  may depend on the replies  $q_1, q_2, \dots, q_{i-1}$ .

**Challenge** Once the adversary decides that Phase 1 is over it outputs two equal length plaintexts  $M_0, M_1 \in \mathcal{M}$  and two identities,  $ID_A$  and  $ID_B$ , on which it wishes to be challenged. The only constraint is that ID did not appear in any extraction query in Phase 1. The challenger pick a random bit  $b \in \{0, 1\}$  and runs  $\text{Extract}(ID_A)$  followed by  $\text{Encrypt}(S_A, ID_B, M_b)$ . It returns the resulting ciphertext  $C^*$  to the adversary.

**Phase 2** During this phase, the adversary may make more queries  $q_{m+1}, \dots, q_n$  of the types described in Phase 1 with the restriction below.

- The Extraction query  $ID_A$  and  $ID_B$  are not permitted.
- The Decryption query  $(ID_A, ID_B, C^*)$  is not permitted

These queries may be asked adaptively in Phase 2.

**Guess** Finally,  $\mathcal{A}$  outputs a bit  $b' \in \{0, 1\}$  and wins the game if  $b = b'$ .

We refer to such an adversary  $\mathcal{A}$  as an IND-CCA attacker. We define the advantage of  $\mathcal{A}$  to be  $\text{Adv}(\mathcal{A}) = |\Pr[b = b'] - \frac{1}{2}|$ . The probability is over the random bits used by the challenger and the adversary.

### 2.2.2 Unforgeability

We say that a scheme is secure against ciphertext forgery if no polynomially-bounded adversary has a non-negligible advantage in the following game.

**Setup** The challenger takes a security parameter  $k$  and runs the Setup algorithm to obtain parameters and a master key  $s$ . It gives the adversary parameters with the value  $s$  such that  $P_{pub} = sP$ .

**Attack** During this phase the adversary makes the queries described below to the challenger.

*Extraction query of the form  $ID_I$*  On receiving such a query the challenger runs  $\text{Extract}(ID_I)$  and responds with  $S_I = x_I d_I$ , where  $d_I$  is a long-term private key generated by the PKG.

*Encryption query of the form  $(ID_I, ID_J, M)$*  On receiving such a query the challenger runs  $\text{Extract}(ID_I)$  followed by  $\text{Encrypt}(S_I, ID_J, M)$ . The response is the resulting ciphertext.

*Decryption query of the form  $(ID_I, ID_J, C)$*  On receiving such a query the challenger runs  $\text{Extract}(ID_J)$  followed by  $\text{Decrypt}(ID_I, S_J, C)$ . The

response is the resulting plaintext  $\mathcal{M}$ . (Sometimes the adversary is notified that the issued ciphertext is invalid.)

**Forge** The adversary attempts to output any valid ciphertext  $C$  from a sender  $A$  to a receiver  $B$ , provided it has not queried the private keys of  $A$  and  $B$  in the previous step. The adversary wins if the ciphertext is valid.

We call such an adversary an EUF-CMA attacker.

### 3 An Authenticated certificateless encryption scheme

Our scheme can be naturally divided four distinct algorithms : Setup, Key Extraction, Encrypt, Decrypt

**Setup:** Given a security parameter  $k$ , the algorithm works as follows:

(1) Run  $\mathcal{IG}$  on input  $k$  to generate a prime  $q$ , two groups  $\mathbb{G}_1, \mathbb{G}_2$  of order  $q$ , and an admissible bilinear map  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ . Choose an arbitrary generator  $P \in \mathbb{G}_1$ .

(2) Pick a random  $s \in \mathbf{Z}_q^*$  and set  $P_{pub} = sP$ .

(3) Choose cryptographic hash functions  $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$ ,  $H_2 : \mathbb{G}_1^* \rightarrow \{0, 1\}^n$ ,  $H_3 : \{0, 1\}^n \times \mathbb{G}_2 \rightarrow \{0, 1\}^n$ ,  $H_4 : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbf{Z}_q^*$  and  $H_5 : \{0, 1\}^n \rightarrow \{0, 1\}^n$ .

Then, output the system parameters  $\langle \mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P_{pub}, H_1, H_2, H_3, H_4, H_5 \rangle$  and the master key  $s$ . The message space is  $\mathcal{M} = \{0, 1\}^n$ . The ciphertext space is  $\mathcal{C} = \mathbb{G}_1 \times \{0, 1\}^n \times \{0, 1\}^n$ .

**Key Extraction :** For a given string  $ID \in \{0, 1\}^*$ , the algorithm does ;

(1) Compute  $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$ .

(2) Pick a random  $x_{ID} \in \mathbf{Z}_q^*$ , set public keys  $X_{ID} = x_{ID}P$  and  $Y_{ID} = x_{ID}Q_{ID}$ .

(3) Set the private keys  $d_{ID}$  to be  $d_{ID} = sQ_{ID}$  and then set  $S_{ID} = x_{ID}d_{ID} = x_{ID}sQ_{ID}$ , where  $s$  is the master key.

**Encrypt :** To encrypt  $M \in \mathcal{M}$ , do the following ;

(1) Choose a random  $\sigma \in \{0, 1\}^n$ .

(2) Set  $r = H_4(\sigma, M)$ .

(3) Compute  $x_A X_B = T$ .

(4) Set the ciphertext to be  $C = \langle rQ_A, \sigma \oplus H_3(H_2(T), \hat{e}(d_A, Y_B)^r), M \oplus H_5(\sigma) \rangle$ , where  $Y_B = x_B Q_B$  is the public key of the receiver.

**Decrypt :** Let  $C = \langle U, V, W \rangle \in \mathcal{C}$ . To decrypt this ciphertext using the private key  $S_B = x_B d_B$ , perform the followings.

(1) Compute  $x_B X_A = T$

(2) Compute  $V \oplus H_3(H_2(T), \hat{e}(U, S_B)) = \sigma$ , where  $S_B = x_B d_B$ .

(3) Compute  $W \oplus H_5(\sigma) = M$

- (4) Set  $r = H_4(\sigma, M)$  and test if  $U = rQ_A$ . If not, reject the ciphertext.  
(5) Output  $M$  as the decryption of  $C$ .

The consistency is easy to verify by the bilinearity. We have  $\hat{e}(d_A, Y_B)^r = \hat{e}(sQ_A, x_BQ_B)^r = \hat{e}(rQ_A, x_BQ_B)^s = \hat{e}(rQ_A, x_BsQ_B) = \hat{e}(U, x_Bd_B) = \hat{e}(U, S_B)$ .

The receiver can be convinced of the origin of the encrypted message by checking if the condition  $rQ_A = U$  holds. Even if the received message would be encrypted under the wrong public key, the receiver could detect the error by testing the final condition.

## 4 Security Analysis of our Scheme

### 4.1 Proof of Integrity

The following theorem shows that our scheme is secure against ciphertext forgery without key escrow, assuming the BDH problem on  $\mathbb{G}_1$  and  $\mathbb{G}_2$  is hard and the CDH problem on  $\mathbb{G}_1$  is hard.

**Theorem 1.** *Let the hash functions  $H_1, H_2, H_3, H_4$  and  $H_5$  be random oracles. Then our scheme is a ciphertext-unforgeable public key encryption assuming the BDHP and the CDHP are hard in groups generated by  $\mathcal{IG}$ . Concretely, suppose  $\mathcal{A}$  is a polynomially-bounded adversary that can forge a ciphertext with advantage  $\epsilon$  and makes at most  $q_E$  extraction queries and at most  $q_{H_1}, q_{H_2}, q_{H_3}$  queries to the hash functions  $H_1, H_2$  and  $H_3$  respectively. Then there exists a polynomially bounded algorithm  $\mathcal{B}$  that solves the BDHP and the CDHP with advantage  $\epsilon / \binom{q_{H_1}}{2} q_D$ .*

*Proof.* Algorithm  $\mathcal{B}$  has as input random and uniformly distributed instances  $(P, aP, bP, cP)$ ,  $(P, xP, yP)$  of the BDHP and CDHP respectively. For finding the value  $\hat{e}(P, P)^{abc}$  and  $xyP$  with  $\mathcal{A}$ 's assistance,  $\mathcal{B}$  has control over the hash functions  $H_1, H_2$  and  $H_3$ . To respond to these hash queries,  $\mathcal{B}$  maintains a list  $L_{H_1}$  that stores information on  $H_1$ -queries, a list  $L_{H_2}$  that stores information on  $H_2$ -queries and a list  $L_{H_3}$  that stores information on  $H_3$ -queries. All lists are initially empty. For simplicity, we assume that all  $H_1$ -queries are distinct (as replies can be cached) and that any query involving an  $ID_A$  is preceded by the  $H_1$ -query for  $ID_A$ . There are several assumptions we may make out  $\mathcal{A}$ 's behavior when interacting with the decryption oracle.

- Before  $\mathcal{A}$  gives its guess,  $\mathcal{A}$  issues a decryption query on it.
- $\mathcal{A}$  does not issue decryption queries on ciphertexts it has received from the encryption oracle or ciphertexts it can compute because it has previously asked for the private key of the sender or receiver.

- Given the above assumptions, we may assume that after every decryption query on a ciphertext, if the answer is a plaintext (i.e. the ciphertext it queried is valid) then  $\mathcal{A}$  stops and outputs this ciphertext.

$\mathcal{B}$  works by interacting with  $\mathcal{A}$  as follows.

**Setup :** At the beginning of the game,  $\mathcal{B}$  gives  $\mathcal{A}$  the system parameters  $\langle \mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P_{pub}, H_1, H_2, H_3, H_4, H_5 \rangle$  with the value  $s$  such that  $P_{pub} = sP$ .

**$H_1$ -queries :**  $\mathcal{B}$  chooses two random numbers  $I, J$  between 1 and  $q_{H_1}$  with  $I \neq J$ . When  $\mathcal{A}$  asks a polynomially bounded number of  $H_1$ -queries on identities of his choice.  $\mathcal{B}$  responds as follows.

(i) At the  $I$ th  $H_1$ -queries,  $\mathcal{B}$  answers  $b_I Q_I$ , where  $Q_I$  is an arbitrary public value. Precisely, if  $ID_A$  does not already appear on the list and  $ID_A$  is the  $I$ th distinct  $H_1$ -query made by  $\mathcal{A}$ , then  $\mathcal{B}$  chooses  $b_I \in \mathbf{Z}_q^*$ , adds  $\langle ID_I, b_I Q_I, b_I, \perp \rangle$  to the list  $L_{H_1}$  and answers  $H_1(ID_I) = b_I Q_I$ .

(ii) At the  $J$ th  $H_1$ -query,  $\mathcal{B}$  answers  $b_J Q_J$ , where  $Q_J$  is an arbitrary public value. Precisely, if  $ID_A$  does not already appear on the list and  $ID_A$  is the  $J$ th distinct  $H_1$ -query made by  $\mathcal{A}$  then  $\mathcal{B}$  chooses  $b_J \in \mathbf{Z}_q^*$ , adds  $\langle ID_J, b_J Q_J, b_J, \perp \rangle$  to the list  $L_{H_1}$  and answers  $H_1(ID_J) = b_J Q_J$ .

(iii) For  $H_1(ID_e)$  where  $e \neq I, J$ ,  $\mathcal{B}$  chooses  $b_e, \alpha_e \in \mathbf{Z}_q^*$  and adds  $\langle ID_e, b_e P, b_e, \alpha_e \rangle$  to the list  $L_{H_1}$  and answers  $H_1(ID_e) = b_e P$ .

**$H_2$ -queries :** A  $H_2$ -query on  $ID_A, ID_B$  is handled as follows ;

(i) If  $ID_A$  and  $ID_B$  are not the identities  $ID_I$  and  $ID_J$  then  $\mathcal{B}$  computes  $\alpha_A \alpha_B P$  adds  $\langle ID_A, ID_B, \alpha_A \alpha_B P, h_2 \rangle$  to the list  $L_{H_2}$  and answers  $h_2$ .

(ii) In the case  $ID_A$  and  $ID_B$  are the identities  $ID_I$  and  $ID_J$ ,  $\mathcal{B}$  chooses  $z \in \mathbf{Z}_q^*$ , adds  $\langle ID_I, ID_J, zP, h_2 \rangle$  to the list  $L_{H_2}$  and answers  $h_2 = H_2(zP)$ .

**$H_3$ -queries :**  $\mathcal{A}$  can issue a  $H_3$ -query request for  $(h_2, U, ID_A, ID_B)$  at any time.  $\mathcal{B}$  runs the  $H_3$ -simulation algorithm to respond  $\mathcal{A}$ 's query as follows.

(i) In the case of  $ID_A = ID_I$ , and  $ID_B = ID_J$ ,  $L_{H_2}$  is examined for an entity of the form  $\langle ID_A, ID_B, \alpha P, h_2 \rangle$  for some  $\alpha$ .

- If such entities are found,  $L_{H_2}$  must contain  $\langle ID_I, ID_J, zP, h_2 \rangle$ . Now  $\mathcal{B}$  chooses  $d^* \in \mathbb{G}_1^*$  randomly, computes  $\hat{e}(U, d^*) = w$  and  $h_3 = H_3(h_2, w)$ , adds the tuple  $\langle ID_I, ID_J, U, (h_2, w), h_3 \rangle$  to the list  $L_{H_3}$ , and then answers  $h_3$ .

- Otherwise,  $\mathcal{B}$  chooses a random  $z' \in \mathbf{Z}_q^*$  and then adds  $\langle ID_I, ID_J, z'P, h_2' \rangle$  to the list  $L_{H_2}$ . Similarly,  $\mathcal{B}$  repeats the remaining process with the new tuple in the  $L_{H_2}$ -list, until obtaining a tuple  $\langle ID_I, ID_J, U, (h_2', w'), h_3' \rangle$ .

(ii) In case  $ID_A \neq ID_I$ ,  $ID_B \neq ID_J$ ,  $\mathcal{B}$  searches a tuple  $\langle ID_A, ID_B, \alpha P, h_2 \rangle$  for some  $\alpha$  in the list  $L_{H_2}$ .



- If such a tuple is found,  $\mathcal{B}$  executes the same process in case (i).  $\mathcal{B}$  computes  $w'' = \hat{e}(U, \alpha_B d_B)$ .  $\mathcal{B}$  could obtain  $\alpha_B d_B = \alpha_B s b_B P$  from the  $L_{H_1}$ -list because  $ID_B \neq ID_J$ . He puts the tuple  $\langle ID_A, ID_B, U, (h_2, w''), h_3'' \rangle$  in the list  $L_{H_3}$  and answers  $h_3''$ .

- Otherwise,  $\mathcal{B}$  chooses a random  $z^* \in \mathbf{Z}_q^*$ , adds  $\langle ID_A, ID_B, z^* P, h_2^* \rangle$  to the  $L_{H_2}$ -list and computes  $w^* = \hat{e}(U, \alpha_B^* d_B)$ . With  $(h_2^*, w^*)$ ,  $\mathcal{B}$  simulates the  $H_3$ -oracle and then obtains  $h_3^* = H_3(h_2^*, w^*)$ . It adds  $\langle ID_A, ID_B, U, (h_2^*, w^*), h_3^* \rangle$  to the list  $L_{H_3}$  and answers  $h_3^*$ .

Key extraction query : When  $\mathcal{A}$  asks a key extraction query on  $ID_B$ ,

(i) If  $ID_A = ID_I$  or  $ID_J$ , then  $\mathcal{B}$  fails and stops.

(ii) If  $ID_A \neq ID_I, ID_J$ , then the list  $L_{H_1}$  must contain  $\langle ID_A, b_A P, b_A, \alpha_A \rangle$ .

The decryption key corresponding to  $ID_A$  is  $\alpha_{As} Q_A = \alpha_A s b_A P = \alpha_A b_{As} P$ .

It is computed by  $\mathcal{B}$  and returned to  $\mathcal{A}$ .

Encryption query : At any time,  $\mathcal{A}$  can perform Encrypt query for a plaintext  $M$  and identities  $ID_A$  and  $ID_B$ .

(i) If  $ID_A = ID_I$  and  $ID_B = ID_J$ ,  $\mathcal{B}$  chooses random values  $r \in \mathbf{Z}_q^*$ ,  $\sigma \in \{0, 1\}^n$ ,  $\alpha_B \in \mathbf{Z}_q^*$ , computes  $U' = r Q_A = r b_I Q_I, V' = \sigma \oplus H_3(H_2(zP), \hat{e}(U', \alpha_B s b_J Q_J)), W' = M \oplus H_5(\sigma)$  and then answers  $C' = \langle U', V', W' \rangle$ .

(ii) If  $ID_A \neq ID_I, ID_B \neq ID_J$ ,  $\mathcal{B}$  computes the private key corresponding to  $ID_A$ . So the ciphertext is computed as described by the PKC algorithm.

Decryption query : Suppose  $\mathcal{A}$  issues an decryption query for a ciphertext  $C = \langle U, V, W \rangle$  between identities  $ID_A$  and  $ID_B$ .

(i) If  $ID_A = ID_I, ID_B = ID_J$ ,  $L_{H_3}$ -list is examined for an entry of the form  $\langle ID_I, ID_J, U, (h_2, w), h_3 \rangle$ . If such an entry is present,  $p = (h_2, w)$  is added to the list  $L_p$ .  $\mathcal{A}$  is notified that  $C$  is invalid, even if  $C$  is valid.

(ii) If  $ID_A \neq ID_I, ID_B \neq ID_J$ , the list  $L_{H_3}$  must contain the entry  $\langle ID_A, ID_B, U, (h_2, w''), h_3'' \rangle$  and so  $\alpha_B s b_B P$  is a decryption key for  $ID_B$ . Then the ciphertext is decrypted as outlined in the description of the PKC algorithm. If it is valid, the plaintext is given to  $\mathcal{A}$  (and  $\mathcal{A}$  wins).

Eventually,  $\mathcal{A}$  terminates. Any output is ignored. Now if  $L_p$  is empty, then  $\mathcal{B}$  fails. Otherwise  $\mathcal{B}$  outputs a random element of  $L_p$ .

*Analysis.* The probability that  $\mathcal{A}$  never issues a key extraction query on one of the guessed ID is at least  $1/\binom{q_{H_1}}{2}$ . (We call any identity that the asked ID is equal to one of values  $ID_I, ID_J$  a *guessed identity*.) If  $\mathcal{A}$  has submitted a valid ciphertext then with a probability greater than  $1/\binom{q_{H_1}}{2}$ ,  $\mathcal{A}$  has successfully forged as ciphertext between the guessed identities (but is returned that the ciphertext is invalid). If  $p = (H_2(xyP), \hat{e}(P, P)^{abc})$  is not in the  $L_p$ -list then  $\mathcal{A}$ 's view is independent of a correct forgery. Hence

the probability that  $\mathcal{A}$  queries  $H_3(p)$  is at least  $\epsilon$ . If this happens then  $\mathcal{B}$  cannot fail and then outputs the correct value with probability at least  $\frac{1}{q_D}$ . We then have  $Adv(\mathcal{B}) \geq \epsilon / \binom{q_{H_1}}{2} q_D$ .  $\square$

## 4.2 Proof of Security for Message Confidentiality

The security of our scheme relies on the intractability of the BDHP and the CDHP. We can state a theorem similar to Theorem 1.

**Theorem 2.** *Let the hash functions  $H_1, H_2, H_3, H_4$  and  $H_5$  be random oracles. We assume our scheme is ciphertext-unforgeable. Then our scheme is a chosen ciphertext secure public key encryption (IND-CCA) assuming the BDHP and the CDHP are hard in groups generated by  $\mathcal{IG}$ . Concretely, suppose  $\mathcal{A}$  is a polynomially bounded IND-CCA adversary with advantage  $\epsilon$  and makes at most  $q_E$  extraction queries, at most  $q_D$  decryption queries and at most  $q_{H_1}, q_{H_2}, q_{H_3}$  queries to the hash functions  $H_1, H_2$  and  $H_3$  respectively, then there exists a polynomially bounded algorithm  $\mathcal{B}$  that solves the BDHP and the CDHP with advantage  $\epsilon / q_{H_2} \binom{q_{H_1}}{2}$ .*

*Proof.* The proof follows the similar steps to the proof of Theorem 1, but differs in the decryption query: since we assume our scheme is ciphertext unforgeable, the decryption oracle's operation must be changed.  $H_1, H_2$  and  $H_3$  hash queries are treated by  $\mathcal{B}$  as in the proof of Theorem 1. To simulate Encryption and Key extraction queries by  $\mathcal{A}$ ,  $\mathcal{B}$  acts exactly as in the proof of Theorem 1. So we only make mention of the decryption queries.

*Phase 1 :* Whenever  $\mathcal{A}$  issues a decryption query, it is notified that the given ciphertext is invalid. By the hypothesis of ciphertext-unforgeability,  $\mathcal{A}$  cannot distinguish between this simulation of a decryption oracle and a real one.

*Challenge :* After a polynomially bounded number of queries,  $\mathcal{A}$  chooses a pair of identities on which he wishes to be challenged. When  $\mathcal{A}$  produces his two plaintexts  $M_0, M_1$  and  $ID_A, ID_B$ ,  $\mathcal{B}$  responds as follows.

- (i) If queried identities are not guessed IDs then  $\mathcal{B}$  fails and stops.
- (ii) Otherwise, the ciphertext is computed as described by the IBE scheme for any random values  $r \in \mathbf{Z}_q^*, \sigma \in \{0, 1\}^n, M_b \in \{M_0, M_1\}$ .

$\mathcal{B}$  answers the challenge  $C = \langle U, V, W \rangle$ .

*Phase 2 :* Key extraction, Encryption, Decryption query ;  $\mathcal{B}$  responds to these queries in the same way it did in the phase 1 of Theorem 1 (except decryption query). But the usual restrictions on  $\mathcal{A}$ 's behavior apply in this phase.

- If  $\mathcal{A}$  asks the private keys of  $ID_I$  or  $ID_J$  before choosing his target identities,  $\mathcal{B}$  fails because he is unable to answer the question.
- If  $\mathcal{A}$  actually chooses to be challenged on  $ID_I$  and  $ID_J$  then he cannot ask the key extraction query for  $ID_I$  or  $ID_J$ 's.
- $\mathcal{A}$  cannot make a decryption query on the challenge ciphertext for the combination of the challenge identities and involving public keys that were used to encrypt  $M_b$ .

*Guess* : Eventually,  $\mathcal{A}$  outputs its guess  $b'$  for  $b$  and wins if  $b = b'$ . Now if  $L_p$  is empty then  $\mathcal{B}$  fails. Otherwise,  $\mathcal{B}$  outputs a random element of  $L_p$ .

*Analysis*. We know that  $\mathcal{B}$  fails if  $\mathcal{A}$  asks the private key associated to the guessed identity during the simulation. We also know that there are  $\binom{q_{H_1}}{2}$  pairs of identities, at least one of them will never be the subject of a key extraction query from  $\mathcal{A}$ . Then, with the probability at least  $1/\binom{q_{H_1}}{2}$ ,  $\mathcal{A}$  does not ask the key extraction of the guessed identities  $ID_I$  and  $ID_J$ . Further, the probability  $\mathcal{A}$ 's challenge identities are the guessed identity pair  $(ID_I, ID_J)$  is  $1/\binom{q_{H_1}}{2}$ . If  $\mathcal{A}$  has never queries  $H_3(p)$  for  $p = (H_2(xyP), \hat{e}(P, P)^{abc})$  then  $\mathcal{A}$ 's view is independent of  $M$ , so in this case  $\mathcal{A}$  is unable to tell that it is in a simulation, and has no advantage. Hence, the probability that  $\mathcal{A}$  queries  $H_3(p)$  is at least  $\epsilon$ . If  $\mathcal{A}$  has queries  $H_3(p)$  then it may be able to distinguish the simulation from the real life, but  $p$  will be cached on  $L_p$ .  $\mathcal{B}$  wins if he guesses the correct element of  $L_p$  to output. But, the size of this list is bounded by  $q_{H_2}$ . Therefore,  $Adv(\mathcal{B}) \geq \epsilon / \binom{q_{H_1}}{2}^2 q_{H_2}$ .

## 5 Comparison

The following table gives a comparison between our scheme and other schemes in terms of efficiency and security properties. Security is indicated as follows: Authentication, without key Escrow, ciphertext Unforgeability, and message Confidentiality.

scheme	# pairings	# multi	# expn	Authen.	Escrow	Unforge	Conf
BF [BF1,?]	2	1	1	X	X	X	O
L [L]	2	0	0	O	X	O	O
AP [AP]	4	1	1	O(half)*	O	O	O
our scheme	2	3	1	O**	O	O	O

\* The scheme satisfies only unilateral authentication.

\*\* Two real-time communicating parties mutually assure each other's identity.

## 6 Conclusions

In this paper, we proposed an authenticated public key encryption scheme. We provided proofs of confidentiality and existential unforgeability under the Bilinear Diffie-Hellman and the Computational Diffie-Hellman assumptions.

The scheme presented in [AP] is somewhat similar to our construction. However, our scheme satisfies mutual authentication, while Al-Riyami and Paterson's scheme provided only unilateral authentication. Moreover, two communicating parties in our model perform encryption/decryption by using a Diffie-Hellman shared secret from their ephemeral contribution.

## References

- [AP] S.S.Al-Riyami, K.G.Paterson. Certificateless Public Key Cryptography. In Proc. Asiacrypt'03, LNCS 2784, Springer Verlag, Lecture Notes in Computer Science series, 2003.
- [BF1] D.Boneh and M.Franklin. Identity-based encryption from the weil pairing, In Proc. Crypto '01, LNCS 2139, pages 213-229, 2001. See [3] for the full version.
- [BF2] D.Boneh and M.Franklin. Identity-based encryption from the weil pairing, SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003.
- [CC] J.C.Cha and J.H.Cheon. An identity-based signature from gap Diffie-Hellman group, Public Key Cryptography 2003: 18-30.
- [CHMSS] L.Chen, K.Harrison, A.Moss, D.Soldera, and N.P.Smart. Certification of public keys within an identity based system. In A.H.Chan and V.D.Gligor, editors, Information Security, 5th International Conference, ISC, volume 2433 of LNCS, pages 322-333. Springer-Verlag, 2002.
- [CK] L.Chen and C.Kudla. Identity based authenticated key agreement from pairing. CSFW 2003: 219-233.
- [D] R.Dupont, A.Enge. Provably secure non-interactive key distribution based on pairings, to appear in Discrete Applied Mathematics. Preliminary version in Proceedings of the International Workshop on Coding and Cryptography, Versailles - WCC 2003.
- [FO] E.Fujisaki, T.Okamoto. Secure integration of asymmetric and symmetric encryption schemes, Advances in Cryptology-Crypto'99, LNCS 1666, Springer, pp.537-554, 1999.
- [PG] P.Gemmel. An introduction to threshold cryptography, in CryptoBytes, a technical newsletter of RSA Laboratories, Vol.2, No.7, 1997.
- [G] C.Gentry. Certificate-Based Encryption and the Certificate Revocation Problem, In E.Biham, editor, Advances in Cryptology-EUROCRYPT 2003, volume 2656 of LNCS, pages 272-193, Springer-Verlag, 2003.
- [H] F.Hess. Efficient identity based signature schemes based on pairings, to appear in proceedings of SAC '2002. Springer Verlag, Lecture Notes in Computer Science series.

- [LQ] B.Libert and J.J.Quisquater. New identity based signcryption schemes based on pairings, IEEE Information Theory Workshop 2003, Paris, France, or full version in Cryptology ePrint Archive, Report 2003/023, 2003, <http://eprint.iacr.org/>.
- [L] B.Lynn, Authenticated identity-based encryption. Cryptology ePrint Archive, Report 2002/072, 2002, <http://eprint.iacr.org/>.
- [P] K.G.Paterson. ID-based signatures from pairings on elliptic curves, Electronics Letters, Vol. 38 (18) (2002), 1025-1026.
- [Sh] A.Shamir. Identity-based cryptosystems and signature schemes. In Proc. Crypto '84, LNCS 196, pages 47-53, 1984.
- [Sm] N.P.Smart. An identity-based authenticated key agreement protocol based on the Weil pairing. Electronics Letters, Vol 38, pp 630-632, (2002).