Efficient Certificateless Public Key Encryption

Yijuan Shi and Jianhua Li

Department of Electronic and Engineering,
Shanghai Jiao Tong University,
Room 408, Building 1, No. 33, Leshan RD, Shanghai, China
cbzsyj130@sohu.com, lijh888@sjtu.edu.cn

Abstract. Certificateless public key cryptography was introduced to overcome the key escrow limitation of the identity-based cryptography. Most of the existing certificateless public key encryption schemes are based on Boneh and Franklin's identity-based encryption scheme (BF-IBE). In this paper, we construct a new certificateless public key encryption scheme from the efficient SK-IBE which has been proved to be IND-ID-CCA secure [8]. The new scheme is more efficient on computation complexity or published public key information than the existing schemes.

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 [1] 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 [2] presented an efficient and provably secure identity-based encryption scheme (BF-IBE) using the bilinear pairings on elliptic curves.

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 [3] 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 [3,4,5,6] are based on the BF-IBE scheme. In 2003 Sakai and Kasahara [7] proposed another method of constructing identity-based keys, also using pairings, which has the potential to improve performance. Later, Chen and Cheng [8] gave a provably secure identity-based scheme (SK-IBE) using this second construction. In this paper, we propose a new CL-PKE scheme based on the efficient SK-IBE scheme. The new scheme is more efficient on computation or published public key information than the existing schemes.

The paper is organized as follows: First we review the concepts of CL-PKE and its security model. 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 [3].

Definition 1. [3] A CL-PKE scheme is specified by seven algorithm (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) [3].

2.1 Security Model for CL-PKE

Al-Riyami and Patersion presented the full IND-CCA security model for CL-PKE in [3]. The following is the actions that an general adversary $\mathcal{A}$ against a CL-PKE scheme may carry out and discuss how each action should be handled by the challenger C for that adversary.

1. **Extract partial private key of entity A:** Challenger C responds by running algorithm **Partial-Private-Key-Extract** to generate the partial private key D_A for entity A.

2. **Extract private key for entity A:** If A's public key has not been replaced then C can respond by running algorithm **Set-Private-Key** to generate the private key S_A for entity A. But it is unreasonable to expect C to be able to respond to such a query if $\mathcal{A}$ has already replaced A's public key.

3. **Request public key of entity A:** C responds by running algorithm **Set-Public-Key** to generate the public key P_A for entity A (first running **Set-Secret-Value** for A if necessary).

4. **Replace public key of entity A:** The adversary $\mathcal{A}$ can repeatedly replace the public key P_A for any entity A with any value P_0 of its choice. The current value of an entity's public key is used by C in any computations or responses to the adversary's requests.

5. **Decryption query for ciphertext C and entity A:** In the model of [3], adversary can issue a decryption query for any entity and any ciphertext. It is assumed in [3] that C should properly decrypt ciphertexts, even for those entities whose public keys have been replaced. This is a rather strong property for the security model (after all, the challenger may no longer know the correct private key). However, it ensures that the model captures the fact that changing an entity's public key to a value of the adversary's choosing may give that adversary an advantage in breaking the scheme. For further discussion of this feature, see [3].

The IND-CCA security model of [3] distinguishes two types of adversary. A type I adversary $\mathcal{A}_I$ is able to change public keys of entities at will, but does not have access to the *masterkey*. A Type II adversary $\mathcal{A}_{II}$ is equipped with the *masterkey* but is not allowed to replace public keys of entities. This adversary models security against an eavesdropping KGC.

CL-PKE Type I IND-CCA 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. As discussed above, we make several natural restrictions on such a Type I adversary:

1. Adversary $\mathcal{A}_I$ cannot extract the private key for ID_{ch} at any point.

2. Adversary $\mathcal{A}_I$ cannot request the private key for any identifier if the corresponding public key has already been replaced.

3. Adversary $\mathcal{A}_I$ cannot both replace the public key for the challenge identifier ID_{ch} before the challenge phase and extract the partial private key for ID_{ch} in some phase.

4. In Phase 2, $\mathcal{A}_I$ cannot make a decryption query on the challenge ciphertext C^* for the combination (ID_{ch}, P_{ch}) that was used to encrypt M_b.

CL-PKE Type II IND-CCA 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. The restrictions on this type of adversary are:

1. Adversary $\mathcal{A}_{II}$ cannot replace public keys at any point.

2. Adversary $\mathcal{A}_{II}$ cannot extract the private key for ID_{ch} at any point.

3. In Phase 2, $\mathcal{A}_{II}$ cannot make a decryption query on the challenge ciphertext C^* for the combination (ID_{ch}, P_{ch}) that was used to encrypt M_b.

Definition 2. A CL-PKE scheme is said to be IND-CCA secure if no polynomially bounded adversary $\mathcal{A}$ of Type I or Type II has a non-negligible advantage in the following game:

Setup: The challenger C takes a security parameter κ as input and runs the Setup algorithm. It gives $\mathcal{A}$ the resulting system parameters *params*. If $\mathcal{A}$ is of Type I, then C keeps the *masterkey* to himself, otherwise, he gives the *masterkey* to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ issues a sequence of requests described above. These queries may be asked adaptively, but are subject to the rules on adversary behavior defined above.

Challenge Phase: Once $\mathcal{A}$ decides that Phase 1 is over it outputs the challenge identifier ID_{ch} and two equal length plaintexts $M_0, M_1 \in \mathcal{M}$. Again, the adversarial constraints given above apply. C now picks a random bit $b \in \{0, 1\}$ and computes C^*, the encryption of M_b under the current public key P_{ch} for ID_{ch}. Then C^* is delivered to $\mathcal{A}$.

Phase 2: Now $\mathcal{A}$ issues a second sequence of requests as in Phase 1, again subject to the rules on adversary behavior above.

Guess: Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. The adversary wins the game if $b = b'$. We define $\mathcal{A}$'s advantage in this game to be $\mathrm{Adv}(\mathcal{A}) := 2|Pr[b = b'] - 1/2|$.

3 Mathematic Basic

Before presenting the new CL-PKE scheme, we first review a few concepts related to bilinear maps [8].

- G_1, G_2 and G_T are cyclic groups of prime order q.
- P_1 is a generator of G_1 and P_2 is a generator of G_2.

– ψ is an isomorphism from G_2 to G_1 with $\psi(P_2) = P_1$.
– $\hat{e}$ is a map $\hat{e}$: $G_1 \times G_2 \rightarrow G_T$.

The map must have the following properties.

Bilinear: For all $P \in G_1$, all $Q \in G_2$ and all $a, b \in Z_q^*$ we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.

Non-degenerate: $\hat{e}(P_1, P_2) \neq 1$.

Computable: For all $P \in G_1$ and all $Q \in G_2$, $\hat{e}(P, P)$ is computable in polynomial time.

From [11], we can either assume that ψ is efficiently computable or make our security proof relative to some oracle with computes ψ.

In the following, we consider the computational Diffie-Hellman (CDH) problem and the k-bilinear Diffie-Hellman inverse (k-BDHI) problem [8,12].

CDH Assumption: For $a, b \in Z_q^*$, given $P, aP, bP \in G_2^*$, to compute abP is hard.

k-BDHI Assumption: For an integer k and $x \in_R Z_q^*$, $P_2 \in G_2^*$, $P_1 \in \psi(P_2)$, $\hat{e} : G_1 \times G_2 \rightarrow G_T$, given ($P_1$, P_2, xP_2,x^2P_2,...x^kP_2), to compute $\hat{e}(P_1,P_2)^{1/x}$ is hard.

Theorem 1: *k-BDHI problem is not harder than CDH problem.*

Proof: CDH problem ⇒ k-BDHI problem

Given $P_1, P_2, xP_2, ..., x^kP_2$, $k \geq 1$,

set the input of CDH problem to be

$P = P_2, aP = xP_2, bP = xP_2$,

CDH problem outputs

$abP = x^2P_2$.

Then set the input of CDH problem to be

$P = x^2P_2, aP = xP_2 = x^{-1}P, bP = P_2 = x^{-2}P$,

CDH problem outputs

$abP = x^{-3}P$.

Hence, compute

$\hat{e}(P_1, x^{-3}P) = \hat{e}(P_1, x^{-1}P_2) = \hat{e}(P_1, P_2)^{x^{-1}}$.

4 A New CL-PKE Scheme

In this section, we present a new CL-PKE scheme and study its security. The scheme is constructed using the SK-IBE scheme of [8] and a variant of the ElGamal public key encryption [9] strengthened using Fujisaki-Okamoto's transform [10].

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

1. Generate three cyclic groups G_1, G_2 and G_T of prime order q, an isomorphism ψ, and a bilinear pairing map $\hat{e}$: $G_1 \times G_2 \rightarrow G_T$. 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 $g = \hat{e}(P_1, P_2)$.

4. Pick five cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow Z_q^*$ and $H_2 : G_T \rightarrow \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow Z_q^*$, $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$ and $H_5 : G_1 \rightarrow \{0,1\}^n$ for some n.

The message space is $\mathcal{M} = \{0,1\}^n$. The ciphertext space is $\mathcal{C} = G_1^* \times \{0,1\}^n \times \{0,1\}^n$. The system parameters are $params =< q, G_1, G_2, G_T, \hat{e}, n, P_1, P_2, g, P_{pub}, H_1, H_2, H_3, H_4, H_5 >$. The *masterkey* is s.

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

Set-Secret-Value: The algorithm takes as inputs *params* and identifier ID_A . It selects a random $x_A \in Z_q^*$ and outputs x_A as A's secret value.

Set-Private-Key: The algorithm takes an inputs *params*, entity A's partial private key D_A and A's secret value x_A. The output of the algorithm is the pair $S_A =< D_A, x_A >$.

Set-Public-Key: The algorithm takes *params* and entity A's secret value x_A as inputs and constructs A's public key as $P_A = x_A(H_1(ID_A)P_1 + P_{pub}) = x_A Q_A$

Encrypt: To encrypt $M \in \mathcal{M}$ for entity A with identifier ID_A and a public key P_A, perform the following steps:

1. Check that P_A is in G_1^*, if not output $\perp$. This checks the validity of the public key.
2. Compute $Q_A = H_1(ID_A)P_1 + P_{pub}$.
3. Choose a random value $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, M)$.
4. Compute and output the ciphertext:

$$C =< rQ_A, \sigma \oplus H_2(\hat{e}(g)^r) \oplus H_5(rP_A), M \oplus H_4(\sigma) >$$

Decrypt: Suppose $C = (U, V, W)$. To decrypt this ciphertext using the private key $S_A =< D_A, x_A >$:

1. Compute $\sigma' = V \oplus H_2(\hat{e}(U, D_A)) \oplus H_5(x_A U)$.
2. Compute $M' = W \oplus H_4(\sigma')$.
3. Set $r' = H_3(\sigma', M')$ and test if $U = r'(H_1(ID_A)P + P_{pub}) = r'Q_A$ where Q_A can be pre-computed. If not, output $\perp$ and reject the ciphertext. Otherwise, output M' as the decryption of C.

4.1 Security of the New CL-PKE

In order for us to prove the security of the new CL-PKE we need to introduce two PKE schemes: SK-IBE [8] and ElG-PKE.

SK-IBE From [8], we can see that the SK-IBE scheme is an efficient identity-based encryption scheme which consists of the following four algorithms:

Setup: Generate the public parameters $< q, G_1, G_2, G_T, \hat{e}, n, P_1, P_2, g, P_{pub}, H_1, H_2, H_3, H_4 >$ and the *masterkey* s . This parameters are identical to the ones

in the above CL-PKE scheme. The message and ciphertext spaces for SK-IBE are $\mathcal{M} = \{0,1\}^n$ and $\mathcal{C} = G_1^* \times \{0,1\}^n \times \{0,1\}^n$.

Extract: The algorithm takes as input an identifier $ID_A \in \{0,1\}^*$ for entity A, public parameters and the *masterkey* s and returns the private key $D_A = (H_1(ID_A) + s)^{-1} P_2$.

Encrypt: To encrypt $M \in \mathcal{M}$ for entity A with identifier ID_A, perform the following steps:

1. Compute $Q_A = H_1(ID_A)P_1 + P_{pub} \in G_1^*$
2. Choose a random value $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, M)$.
3. Compute and output the ciphertext:

$$C =< rQ_A, \sigma \oplus H_2(\hat{e}(g)^r), M \oplus H_4(\sigma) >$$

Decrypt: Suppose $C = (U, V, W)$. To decrypt this ciphertext using the private key D_A:

1. Compute $\sigma' = V \oplus H_2(U, D_A)$.
2. Compute $M' = W \oplus H_4(\sigma')$.
3. Set $r' = H_3(\sigma', M')$ and test if $U = r'(H_1(ID_A)P + P_{pub}) = r'Q_A$ where Q_A can be pre-computed. If not, output $\perp$ and reject the ciphertext. Otherwise, output M' as the decryption of C.

Theorem 2: *[8] SK-IBE is secure against IND-ID-CCA [2] adversaries provided that $H_i (1 \leq i \leq 4)$ are random oracles and the k-BDHI assumption is sound.*

ElG-PKE ElG-PKE which is similar with the ElG-HybridPub in [13] is specified by four algorithms:

Setup: Given a security parameter κ, generate the public parameters $< q, G_1, G_2, G_T, \hat{e}, n, H_3, H_4, H_5 >$. This parameters are identical to the ones in the above CL-PKE. Pick a random $P \in G_1^*$. The message and ciphertext spaces for ElG-PKE are $\mathcal{M} = \{0,1\}^n$ and $\mathcal{C} = G_1^* \times \{0,1\}^n \times \{0,1\}^n$.

Key-Generation: Choose a random $x \in Z_q^*$ and set $R = xP$. Set the public key K_{pub} to be $< q, G_1, G_2, G_T, \hat{e}, n, P, H_3, H_4, H_5, R >$ and the private key K_{pri} to be x.

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

1. Choose a random value $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, M)$.
2. Compute the ciphertext:

$$C =< rP, \sigma \oplus H_5(rR), M \oplus H_4(\sigma) >.$$

Decrypt: To decrypt the ciphertext $C =< U, V, W > \in \mathcal{C}$ using the private key x, do the follows:

1. Compute $\sigma' = V \oplus H_5(xU)$.
2. Compute $M' = W \oplus H_4(\sigma')$.
3. Set $r' = H_3(\sigma', M')$ and test if $U = r'P$. If not, output $\perp$ and reject the ciphertext. Otherwise, output M' as the decryption of C.

Theorem 3: [13] *ElG-PKE is secure against IND-CCA adversaries provided that* $H_i(3 \leq i \leq 5)$ *are random oracles and the CDH assumption is sound.*

According to the security of the above SK-IBE scheme and the ElG-PKE scheme, we can prove the security of our new CL-PKE scheme formally in the similar method in [13]. For the limited space, we skip the detailed formal proof here and only analyze the security of our CL-PKE scheme heuristically according to the security model in section 2.

Type I adversary $\mathcal{A}_I$: $\mathcal{A}_I$ does not know the *masterkey* 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 A with $P'_A = xQ_A$. If a sender wants to encrypt a message $M \in \mathcal{M}$ for entity A, he computes the CL-PKE ciphertext as:
$C =< rQ_A, \sigma \oplus H_2(\hat{e}(g)^r) \oplus H_5(rP'_A), M \oplus H_4(\sigma) >=< U, V, W >$.
Then $\mathcal{A}_I$ with x can compute
$C' =< U, V \oplus H_5(xU), W >=< rQ_A, \sigma \oplus H_2(\hat{e}(g^r)), M \oplus H_4(\sigma) >$
which is the SK-IBE encryption for the message M. Hence, the security of the CL-PKE can be reduced to the security of the SK-IBE scheme which is based on the hardness of the k-BDHI 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_A for the entity A. If a sender wants to encrypt a message $M \in \mathcal{M}$ for entity A, he computes the CL-PKE ciphertext as:
$C =< rQ_A, \sigma \oplus H_2(\hat{e}(g)^r) \oplus H_5(rP_A), M \oplus H_4(\sigma) >=< U, V, W >$.
Then $\mathcal{A}_I$ with D_A can compute
$C' =< U, V \oplus H_2(\hat{e}(U, D_A)), W >=< rQ_A, \sigma \oplus H_5(rP_A), M \oplus H_4(\sigma) >$
which is the ElG-PKE encryption for the message M with $R = P_A$. Hence the security of the CL-PKE can be reduced to the security of the ElG-PKE which is based on the hardness of the CDH problem.

From theorem 1, we know that the k-BDHI problem is not harder than the CDH problem. Therefore, we say that the security of the new CL-PKE scheme is based on the hardness of the k-BDHI problem.

Theorem 4: *The new CL-PKE scheme is secure against IND-CCA adversaries provided that* $H_i(1 \leq i \leq 5)$ *are random oracles and the k-BDHI assumption is sound.*

5 Performance Analysis

In this section, we will show that our proposed CL-PKE scheme has the best performance, comparing with other existing CL-PKE schemes [3,4,5,6]. All the schemes have three major operations, i.e., Pairing (p), Scalar(s) and Exponentiation (e). Without considering the pre-computation, the properties and per-

formance of the CL-PKE schemes are listed in Table 1, where we compare the schemes on the computation complexity, public key length (Pubkey Len) and the hardness assumption.

We know that pairing computation is more time-consuming than scalar and exponentiation computation [14]. From Table 1 we can see that our new scheme requires no pairing computation in Encrypt and the public key consists of only one element of G_1 rather than two required in AP's scheme I and CC's scheme I. Hence, our scheme is more efficient than the existing CL-PKE schemes.

Table 1. Comparison of the CL-PKE Schemes

Schemes	Encrypt	Decrypt	Pubkey Len	Assumption
AP's scheme I [3]	3p+1s+1e	1p+1s	2	GBDH
CC's scheme I [5]	3p+1s+1e	1p+1s	2	BEQ
AP's scheme II [4]	1p+2s+1e	1p+2s	1	BDH
CC's scheme II [6]	1p+2s+1e	1p+2s	1	BDH
New scheme	3s+1e	1p+2s	1	k-BDHI

6 Conclusions

In this paper, we present an efficient CL-PKE scheme which is constructed by the provable efficient SK-IBE rather than BF-IBE. Based on the security of the SK-IBE scheme and the ElG-PKE scheme, we analyze the security of our scheme heuristically. In fact, we can prove the security of our new CL-PKE scheme formally in the similar method in [13]. Furthermore, our scheme is more efficient than the existing CL-PKE schemes on computation or published public key information.

References

1. Shamir, A.,: Identity based Cryptosystems and Signature Schemes. Advances in Cryptology-CRYPTO'84, Springer-verlag, LNCS 196 (1985) 47–53
2. Boneh, D. and Franklin, M.,: Identity based Encryption from the Weil Pairing. Advances in Cryptology-CRYPTO'2001, Springer-Verlg, LNCS 2139 (2001) 213–229
3. Al-Riyami, S.S. and Paterson, K.G.,: Certificateless Public Key Cryptography. In Advances in Cryptology C ASIACRYPT 2003, Springer-verlag LNCS vol. 2894 (2003) 452-C473
4. Al-Riyami, S.S. and Paterson, K.G.,: CBE from CL-PKE: A Generic Construciton and Efficient Schemes. PKC 2005, LNCS 3386 (2005) 398–415
5. Cheng, Z.H., Comley, R. and Vasiu, L.,: Remove Key Escrow from The Identity-Based Encryption System. TCS@IFIP, Toulouse, France, August 2004. Foundations of Information Technology in the Era of Network and Mobile Computing.

6. Cheng, Z.H. and Comley, R.,: Efficient Certificateless Public Key Encryption. Cryptology ePrint Archive, Report 2005/012
7. Sakai, R. and Kasahara, M.,: ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054
8. Chen, L.Q. and Cheng, Z.H.,: Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme. Cryptology ePrint Archive, Report 2005/226
9. ElGamal, T.,: A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31:469-472, 1985
10. Fujisaki, E. and Okamotom, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. Advances in Cryptology - CRYPTO 1999 Proceedings, Springer-Verlag (1999) 535–554
11. Smart, N. and Vercauteren, F.,: On computable isomorphisms in efficient pairing based systems. Cryptology ePrint Archive, Report 2005/116
12. Boneh, D. and Boyen, X.,: efficient selective-ID secure identity-based encryption without random oracles. In Proceedings of Advances in Cryptology - Eurocrypt 2004, LNCS 3027, Springer-Verlag (2004) 223–238
13. Al-Riyami, S.S.,: Cryptographic schemes based on elliptic curve pairings. Ph.D. thesis, University of London, 2004
14. Barreto, P.S.L.M., Lynn, H.Y. and Scott, M.,: Efficient Algorithms for Pairing-based cryptosystems. Advances in Cryptology-Crypto 2002, LNCS Vol. 2442 (2002) 354–368