Remove Key Escrow from The Identity-Based Encryption System

Zhaohui Cheng@mdx.ac.uk

Abstract

Key escrow is an inherent property in the current proposed Identity-Based Encryption (IBE) systems. However key escrow is not always a good property for all applications. In this paper, we provide a scheme to remove the key escrow from the IBE system proposed by Bonch and Franklin, while at the same time maintain the important properties of IBE. We also present some cryptosystems based on our variant including a signature scheme and an authenticated key agreement. We finally show how to integrate our scheme into a hierarchial identity based public key encryption system.

Keywords: Identity-based encryption, Key escrow

1 Introduction

Since the landmark paper "New directions in cryptography"[5] published in 1976, the public key system has been playing a fundamental role in the modern information security society. To address the threat of a new attack method "woman-in-the-middle", a public key certification system was built up. But the widespread deployment of public key system depends heavily on the certification distribution system.

In an attempt to simplify certification management in a Public Key Center (PKC), in 1984 Shamir [10] first formulated a special public key encryption scheme (Identity-Based Encryption: IBE) in which the public key can be an arbitrary string. In such a scheme there are four algorithms: (1) setup generates global system parameters and a master-key, (2) extract uses the master-key to generate the private key corresponding to an arbitrary public key string $ID \in \{0,1\}^*$, (3) encrypt encrypts messages using the public key ID, and (4) decrypt decrypts messages using the corresponding private key.

Since the problem was posed in 1984 there have been several proposals for IBE schemes. Until recently, however only a few practical schemes have been achieved including Boneh-Franklin's IBE system[2] and Cocks' scheme[4].

Because the entity's public key is the entity's ID, some interesting usages of IBE are naturally introduced, for example ID can include public key expiring time, or ID can differentiate the user's credentials, etc. in compliance with different requirements. One additional property is inherent in the proposed IBE scheme. In Shamir's scheme, the PKC uses the algorithm (2) to generate a private key corresponding to the public ID, so the PKC knows all the entities' private keys. That means the key escrow function is part of an IBE system. As proposed schemes [2],[4] follow Shamir's scheme to produce systems, they also have key escrow function. But key escrow is not always a good property for all types of applications. Once the master-key is exposed, all the entities' private keys are leaked in principle and all the prior communication contents are under threat of exposure. Threshold cryptography [6] can be used to increase the security of the master-key. Gentry and Silverberg [7] present a method in the hierarchical ID-based scheme to restrict the key escrow function in a small area. But the existence of master-key still threat the entities' privacy. Al-Riyami and Paterson [1] first introduce the concept of "Certificateless Public Key Cryptography" (CL-PKC) and remove the key escrow successfully. In this paper, we introduce the "Nickname" concept and present another variant of Boneh-Franklin's IBE system without key escrow property, but with concise concept and efficient computation comparing with the CL-PKC scheme in [1].

The rest of this paper is structured as follows. In section 2, we describe the original Boneh-Franklin's IBE, which is the basis of our variant and introduce the bilinear map which is the basic technique of Boneh-Franklin's scheme. In section 3, we present our scheme which removes the key escrow function. Section 4 is a signature scheme based on our variant. And we provide an authenticated key agreement in section 5. Finally we make a comparison with the CL-PKC scheme in section 6.

2 Boneh-Franklin's IBE Scheme

Boneh-Franklin's IBE scheme is the first efficient and security proved identity-based encryption scheme, which is based on a "bilinear map" $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of large prime order q. The modified Weil and Tate pairings [11] on elliptic curves can be used to build secure bilinear maps. The bilinear map has the following properties:

1. Bilinear: For all $P, Q, R, S \in \mathbb{G}_1$, $\hat{e}(P+Q, R+S) = \hat{e}(P,R)\hat{e}(P,S)$ $\hat{e}(Q,P)\hat{e}(Q,S)$[1].
2. Non-Degenerate: For a given point $Q \in \mathbb{G}_1$, $\hat{e}(Q,R) = 1_{\mathbb{G}_2}$ for all $R \in \mathbb{G}_1$ if and only if $Q = 0_{\mathbb{G}_1}$.

[1] In particular $\hat{e}(sP, tR) = \hat{e}(P,R)^{st}$ for all $P, R \in \mathbb{G}_1$ and $s, t \in \mathbb{Z}_q$

3. Computable: There is an efficient algorithm to compute $\hat{e}(P,Q)$ for any $P, Q \in \mathbb{G}_1$.

The security of the scheme is based on an assumption on the hardness of "Bilinear Diffie-Hellman" (BDH) problem

Assumption 1. *BDH Assumption. Let $\mathcal{G}$ be a BDH parameter generator.*

$$Adv_{\mathcal{G},\mathcal{A}}(k) = Pr[\mathcal{A}(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, aP, bP, cP) = \hat{e}(P,P)^{abc} \mid \begin{array}{l} < q, \mathbb{G}_1, \mathbb{G}_2, \hat{e} > \leftarrow \mathcal{G}(1^k), \\ P \leftarrow \mathbb{G}_1, a, b, c \stackrel{R}{\leftarrow} \mathbb{Z}_q^* \end{array}].$$

For any randomized polynomial time (in k) algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{G},\mathcal{A}}(k)$ is negligible (We say that it is hard to solve this problem).

Boneh-Franklin's IBE scheme also follows the four steps proposed in Shamir's scheme. Here is the description of the scheme in detail.

Setup: Given a security parameter k, the parameter generator follows the steps.

1. generate cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order q and a bilinear pairing map[2] $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Pick a random generator $P \in \mathbb{G}_1$.
2. pick a random $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$.
3. pick cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1^*, H_2 : \mathbb{G}_2 \to \{0,1\}^n, H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q^*, H_4 : \{0,1\}^n \to \{0,1\}^n$ for some integer $n > 0$.

The message space is $\mathcal{M} = \{0,1\}^n$. The ciphertext space is $\mathcal{C} = \mathbb{G}_1^* \times \{0,1\}^n \times \{0,1\}^n$. The system parameters are **params** $=< q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2, H_3, H_4 >$. The **master-key** is $s \in \mathbb{Z}_q^*$.

Extract: Given a string $ID \in \{0,1\}^*$, **params** and **master-key**, the algorithm computes $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$ and $d_{ID} = sQ_{ID}$ and returns d_{ID}.

Encrypt: Given a plaintext $m \in \mathcal{M}$, ID and public parameters **params**,

1. compute $Q_{ID} = H_1(ID)$,
2. pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, m)$,
3. compute $g = \hat{e}(P_{pub}, Q_{ID})$,
4. set the ciphertext to $C =< rP, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma) >$.

Decrypt: Given a ciphertext $< U, V, W > \in \mathcal{C}$, a private key d_{ID} and system parameters **params**

[2] In [2], the concrete IBE uses an admissible map

1. compute $g^{'} = \hat{e}(U, d_{ID})$,
2. compute $\sigma^{'} = V \oplus H_2(g^{'})$,
3. compute $m^{'} = W \oplus H_4(\sigma^{'})$,
4. compute $r^{'} = H_3(\sigma^{'}, m^{'})$, If $U \neq r^{'}P$, reject the ciphertext, else return $m^{'}$ as the plaintext.

The consistency of this scheme follows from the bilinearity of $\hat{e}$ and Boneh and Franklin prove that the scheme is semantically secure against the adaptive chosen ciphtertext attack (IND-IN-CCA) in the random oracle.

3 Our Variant of Boneh-Franklin's IBE system

Our scheme is based on Boneh-Franklin's scheme, but to remove the key escrow function, we introduce another public and private key pair $< N_{ID}, t >$. The private key t is only owned by the entity ID, which can be generated by random selection from $\mathbb{Z}_q^*$. In our scheme encryption and decryption not only depend on the public key ID (in fact it is Q_{ID}) and private key d_{ID}, but also on the other public key N_{ID} and corresponding private key t. We name public keys $< ID, N_{ID} >$ as $< ID, Nickname >$ and private keys $< d_{ID}, t >$ as $< PrKeyL, PrKeyR >$. As only entity ID knows the PrKeyR, the key escrow function in PKC is removed. The effect of introducing $< N_{ID}, t >$ is discussed after the scheme details, and we will see that the publication of $Nickname$ is not a serious new burden for PKC. In the following chapters, we name our system as **V-IBE** and Boneh-Franklin's scheme as **B-IBE** for simplicity.

Our scheme is specified by five algorithms:**Setup**, **Extract**, **Publish**, **Encrypt**, **Decrypt**.

Setup: As in Boneh-Franklin's scheme.

Extract: As the algorithm Extract in Boneh-Franklin's scheme.

Publish: Given a system **params** and an entity ID, select a random $t \in \mathbb{Z}_q^*$, and computes $N_{ID} = (N_1, N_2) = (tP, tP_{pub})$ (The entity can ask the PKC to publish this extra parameter N_{ID} or publish it by itself as a nickname).

Encrypt: Given a plaintext $m \in \mathcal{M}$, the identifier ID and public parameters **params**, and the nickname $N_{ID} = (N_1, N_2)$ corresponding to the ID, follow the steps

1. check that $N_1, N_2 \in \mathbb{G}_1^*$ and that the equality $\hat{e}(N_1, P_{pub}) = \hat{e}(N_2, P)$ holds. If not output $\perp$ and terminate encryption.

2. compute $Q_{ID} = H_1(ID)$,
3. pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, m)$,
4. compute $g = \hat{e}(P_{pub} + N_1, Q_{ID})$,
5. set the ciphertext to $C =< rP, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma) >$.

Decrypt: Given a ciphertext $< U, V, W > \in \mathcal{C}$, the PrKeyL d_{ID} from PKC, the PrkeyR t and system parameters **params**

1. compute $g^{'} = \hat{e}(U, d_{ID} + tQ_{ID})$,
2. compute $\sigma^{'} = V \oplus H_2(g^{'})$,
3. compute $m^{'} = W \oplus H_4(\sigma^{'})$,
4. compute $r^{'} = H_3(\sigma^{'}, m^{'})$, If $U \neq r^{'}P$, reject the ciphertext, else return $m^{'}$ as the plaintext.

First let us verify the consistence.

$$\begin{aligned} g^{'} &= \hat{e}(U, d_{ID} + tQ_{ID}) = \hat{e}(rP, sQ_{ID} + tQ_{ID}) \\ &= \hat{e}(sP, Q_{ID})^r \hat{e}(tP, Q_{ID})^r = \hat{e}(P_{pub} + N_1, Q_{ID})^r \end{aligned}$$

So $\sigma^{'}$ in decryption equals σ in encryption. Thus, applying decryption after encryption produces the original message m.

Based on the BDH assumption, this scheme has some special properties which make it different from the normal public key systems and the existing identity-based encryption schemes.

Claim 1. No more key escrow. *Without knowing private key t, the adversary can't decrypt the message, even with information of the master-key s.*

This claim follows from the Theorem 1 in the following section.

Claim 2. Partial identity-based. *Without knowing d_{ID}, the adversary can't decrypt the message.*

This claim follows from the Theorem 2 in the following section.

Remark 1. Loose binding public key. *The extra public key parameter N_{ID} introduced in our scheme need not be bound strictly (by secure method) to the entity with identity ID. It can be distributed through an unsafe channel as the entity's nickname. Once two persons Alice and Bob want to communicate securely, if Alice wants to send a message to Bob, but doesn't know Bob's nickname, she can ask Bob directly or query PKC. Because of Claim 2, the communication cannot be hijacked by Eve who launches the woman-in-the-middle attack and changes the Bob's nickname with her own*

nickname except that Eve is PKC. This character differentiates our scheme from the normal certification based public key systems. In[1], one easy way is explained to thwart the PKC to impersonate an other entity in the woman-in-the-middle attack. The basic idea is to bind the entity's identifier and its public key by re-defining $Q_A = H_1(ID_A \| N_A)$.

Remark 2. Forward Security of Master Key. *Our scheme introduces an extra public and private key pair* $< N_{ID}, t >$ *and only the entity with* ID *knows the private key* t. *So even if the master key* s *of PKC is leaked, the prior communications with destination to entity* ID *would not be exposed, but the following communication would become vulnerable to the woman-in-the-middle attack.*

4 V-IBE's Security

By defining two types adversaries, which correspond to the adversary with *master-key* and the adversary without *master-key* respectively, we state the security analysis in the following two theorems(we don't give out quantity analysis).

Definition: Type-I Attack

The adversary has the *master-key* and launches the Type-I attack by taking one or more of following actions

1. Query nickname for any entity ID_i.
2. Extract PrKeyL d_i for any entity ID_i. Because the adversary has the master-key, so it can compute PrKeyL for any entity. We still assume the adversary issues Extract to get the PrKeyL from the challenger.
3. Issue decryption query $< ID_i, Nickname_i, C_i >$.
4. Publish nickname for any entity ID_i except ID_{ch}.

If the adversary with the master-key also changes the nickname of some entity N_{ID} on which it wants to be challenged, our scheme can't protect the information encrypted under Q_{ID} and the changed N_{ID}. In traditional public key cryptosystems this attack is either not precluded. So we ignore such kind of attack in our proof.

Theorem 1. *If there exists an Type-I IND-CCA adversary* $\mathcal{A}$ *has non-negligible advantage against V-IBE, then there exists an adversary* $\mathcal{B}$ *has non-negligible advantage against BDHP in the random oracle model.*

Definition: Type-II Attack

The adversary launching Type-II attack can take one or more of the following actions.

1. Query nickname for entity ID_i.
2. Extract PrKeyL d_i for any entity ID_i except ID_{ch}.
3. Issue decryption query $< ID_i, Nikename_i, C_i >$.
4. Publish nickname for any entity ID_i.

The adversary can query private PrKeyL for any entity ID_i except ID_{ch} and can publish nickname for any entity.

Theorem 2. *If there exists an IND-CCA Type-II adversary $\mathcal{A}$ against V-IBE with advantage ϵ, then there exits an adversary $\mathcal{B}$ against BDHP or DHP (Diffie-Hellman Problem) in the random oracle model.*

We don't present the formal proof of the security of the whole scheme in this paper, but show the stubbornness to some possible attacks. Firstly we prove that based on BDH assumption, it is hard for the PKC to compute $g^{'}$ in the decryption, even it knows the master key s. To construct $g^{'}$, the PKC needs to use the available information $(s, P, U = rP, Q_{ID} = aP, N_{ID} = tP)$ to compute $\hat{e}(U, d_{ID} + tQ_{ID}) = \hat{e}(rP, saP + taP) = \hat{e}(P, P)^{ra(s+t)}$.

Lemma 1. *Given $(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, s, P, aP, rP, tP)$, where $a, r, t \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$ and s is a fixed element from $\mathbb{Z}_q^*$, based on BDH assumption, it is hard to compute $\hat{e}(P, P)^{ra(s+t)}$.*

Proof. The proof is straight forward. If an adversary $\mathcal{A}$ can solve the above problem, we can construct an adversary $\mathcal{B}$ using $\mathcal{A}$ as subroutine to solve the BDH problem. Given the challenge (P, aP, bP, cP), $\mathcal{B}$ simply randomly selects an element s from $\mathbb{Z}_q^*$ and passes (s, P, aP, bP, cP) as the challenge to $\mathcal{A}$. Once got the response R from $\mathcal{A}$, $\mathcal{B}$ computes $\hat{e}(aP, bP)^{-s}$ and returns $R \cdot \hat{e}(aP, bP)^{-s}$ as the response to the BDH challenge. If $\mathcal{A}$ wins the game with non-negligible advantage, so is $\mathcal{B}$ because if $R = \hat{e}(P, P)^{ba(s+c)}$, $\mathcal{B}$'s response is $\hat{e}(P, P)^{ba(s+c)}\hat{e}(aP, bP)^{-s} = \hat{e}(P, P)^{abc}$.

Secondly we show that if an adversary without the master key wants to compute $g^{'}$ in the decryption by selecting specific N_1 to find $(s+t)$, it needs to break the DHP. Without the check step, the scheme is obviously insecure. An adversary can randomly select $j \in \mathbb{Z}_q^*$ and set $N_1 = tP = -P_{pub} + jP$ $(s + t = j \mod q)$, so it can compute $g^{'} = \hat{e}(U, Q_{ID})^j$. But with applying the check step, the adversary needs to find $tsP = (j - s)sP$ to pass the check step, which means it needs to compute s^2P. It is easy to prove that given $(\mathbb{G}_1, q, P, sP)$ to compute s^2P is a DH problem in group $\mathbb{G}_1$ [9].

5 A Signature Scheme Based on Our Variant

We describe a public key signature (PKS) scheme based on a provably secure ID-PKC signature scheme in [8] and our variant.

The PKS scheme can be specified by algorithms: Setup, Extract, Publish, Sign and Verify.

Setup: Given a security parameter k, the parameter generator follows the steps.

1. generate cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order q and a bilinear pairing map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Pick a random generator $P \in \mathbb{G}_1$.
2. pick a random $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$.
3. pick cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*, H : \{0,1\}^* \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$

Extract: Given a string $ID \in \{0,1\}^*$, **params** and **master-key**, the algorithm computes $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$ and $d_{ID} = sQ_{ID}$ and returns d_{ID}.

Publish: Given a system **params** and an entity ID, select a random $t \in \mathbb{Z}_q^*$, and computes $N_{ID} = (N_1, N_2) = (tP, tP_{pub})$.

Sign: To sign a message $m \in \mathcal{M}$ using the private key $< d_{ID}, t >$, follow the steps:

1. choose an arbitrary point $P_1 \in \mathbb{G}_1^*$, pick a random integer $k \in \mathbb{Z}_q^*$.
2. compute $r = \hat{e}(kP_1, P)$.
3. compute $v = H(m, r)$.
4. compute $U = v(d_{ID} + tQ_{ID}) + kP_1$.
5. output as the signature $< U, v >$.

Verify: To verify a signature $< U, v >$ on a message $m \in \mathcal{M}$ for identity $< ID, N_{ID} >$.

1. check that $N_1, N_2 \in \mathbb{G}_1^*$ and that the equality $\hat{e}(N_1, P_{pub}) = \hat{e}(N_2, P)$ holds. If not output $\perp$ and terminate verification.
2. compute $Q_{ID} = H_1(ID)$,
3. compute $r' = \hat{e}(U, P)\hat{e}(Q_{ID}, -P_{pub} - N_1)^v$
4. accept the signature if and only if $v = H(m, r')$.

The consistence easily follows from

$$\begin{aligned} r' &= \hat{e}(U,P)\hat{e}(Q_{ID},-P_{pub}-N_1)^v \\ &= \hat{e}(vd_{ID}+vtQ_{ID}+kP_1,P)\hat{e}(vQ_{ID},-sP)\hat{e}(vQ_{ID},-N_{ID}) \\ &= \hat{e}(vsQ_{ID},P)\hat{e}(vtQ_{ID},P)\hat{e}(kP_1,P)\hat{e}(vsQ_{ID},-P)\hat{e}(vtQ_{ID},-P) \\ &= \hat{e}(kP_1,P) \end{aligned}$$

6 An Authenticated Key Agreement Protocol

The following is a two-party key agreement which extends the protocol 1 in [3], which is vulnerable to the standard woman-in-the-middle attack launched by PKC.

$$\begin{aligned} A \to B: &\quad xP, N_{ID}^A = (N_1^A, N_2^A) = (aP, aP_{pub}) \qquad (1) \\ B \to A: &\quad yP, N_{ID}^B = (N_1^B, N_2^B) = (bP, bP_{pub}) \qquad (2) \end{aligned}$$

Upon the completion of message exchanges, A and B first check the exchanged nicknames, after that A computes $K_A = \hat{e}(Q_{ID}^B, P_{pub}+N_1^B)^x\hat{e}(d_{ID}^A + aQ_{ID}^A, yP)$, and B computes $K_B = \hat{e}(Q_{ID}^A, P_{pub}+N_1^A)^y\hat{e}(d_{ID}^B + bQ_{ID}^B, xP)$ respectively. It is easy to see that $K = K_A = K_B$ is shared between A and B.

$$\begin{aligned} K_A &= \hat{e}(Q_{ID}^B, sP+bP)^x\hat{e}(sQ_{ID}^A + aQ_{ID}^A, yP) \\ &= \hat{e}(sQ_{ID}^B + bQ_{ID}^B, xP)\hat{e}(Q_{ID}^A, sP+aP)^y \\ &= K_B \end{aligned}$$

To enable forward security, A and B can use $H(K\|xyP)$ as the shared key where H is a proper hash function.

This protocol is still vulnerable to the woman-in-the-middle launched by the PKC which replaces the nicknames in the two messages with its own selections. We can use the same method mentioned in Section 3 to thwart such attack.

7 Hierarchical PKE

Gentry and Silverberg [7] introduced a totally collusion-resistant hierarchical ID-based infrastructure for encryption and signature. We integrate our scheme into this hierarchical system to eliminate all kinds of key escrow to any ancestor of an entity. In [7], every entity is located in one level of a hierarchical system. Except for the root entity, every entity is identified by an ID-tuple which identifies every ancestor along the path to the root. The major steps are identical to the steps in [7].

Root Setup: Given a security parameter k, the parameter generator follows the steps.

1. generate cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order q and a bilinear pairing map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Pick a random generator $P_0 \in \mathbb{G}_1$.

2. pick a random $s_0 \in \mathbb{Z}_q^*$ and compute $Q_0 = s_0 P_0$.

3. pick cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*, H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ for some integer $n > 0$.

Low-lever Setup: Entity $E_t \in Level_t$ picks a random $s_t \in \mathbb{Z}_q^*$, which it keeps secret.

Extraction: Let E_t be an entity in $Level_t$ with ID-tuple $(ID_1, \ldots, ID_t)$, where $(ID_1, \ldots, ID_i)$ for $1 \leq i \leq t$ is the ID-tuple of E_t 's ancestor at $Level_i$. Follow the steps:

1. compute $P_t = H_1(ID_1 \| ID_2 \| \ldots \| ID_t) \in \mathbb{G}_1$.

2. set E_t's secret point $S_t = S_{t-1} + s_{t-1} P_t = \sum_{i=1}^{t} s_{i-1} P_i$

3. set $Q_i = s_i P_0$ for $1 \leq i \leq t-1$.

Publish: For D_t, select a random b_t and compute the nickname $N_t = (N_1^t, N_2^t) = (b_t P_0, b_t Q_0)$.

Encryption: To encrypt $m \in \mathcal{M}$ with the ID-tuple $(ID_1, \ldots, ID_t)$, do the following:

1. for each $1 \leq i \leq t$, check that $N_1^i, N_2^i \in \mathbb{G}_1^*$ and that the equality $\hat{e}(N_1^i, Q_0) = \hat{e}(N_2^i, P_0)$ holds. If not output $\perp$ and terminate encryption.

2. compute $P_t = H_1(ID_1 \| ID_2 \| \ldots \| ID_i) \in \mathbb{G}_1$ for $1 \leq i \leq t$.

3. choose random $r \in \mathbb{Z}_q^*$.

4. compute ciphertext $C =< U_0, U_2, \ldots, U_t, V >=< rP_0, rP_2, \ldots, rP_t, m \oplus H_2(g^r) >$, where $g = \hat{e}(Q_0 + N_1^t, P_1) = \hat{e}(s_0 P_0, P_1) \hat{e}(b_t P_0, P_1)$.

Decryption: To decrypt the ciphertext $C =< U_0, U_2, \ldots, U_t, V > \in \mathcal{C}_t$ for a level t entity with ID-tuple $(ID_1, ID_2, \ldots, ID_t)$, follow the steps:

1. $g' = \frac{\hat{e}(U_0, S_t + b_t P_1)}{\prod_{i=2}^{t} \hat{e}(Q_{i-1}, U_i)} = \hat{e}(rP_0, s_0 P_1 + b_t P_1) = \hat{e}(s_0 P_0, P_1)^r \hat{e}(b_t P_0, P_1)^r$.

2. compute $m' = V \oplus H_2(g')$ as the plaintext.

8 Comparison with CL-PKC

First we have shown that all the cryptosystems based on CL-PKC can be realized using our variant. In our scheme the public key is still partial identity-based, which is made of two parts $< ID, Nickname >$. Whereas in

CL-PKC, the public key also consists of two parts, but the public key in the CL-PKC is not directly associated with the entity ID and system **params**.

Our scheme is slightly slower than CL-PKC, because our scheme needs an extra point addition operation. But the point addition is very fast comparing with the pairing computation or scalar operation. The following table compares complexity of two schemes (P for pairing computation, S for scalar operation and E for exponentiation). We ignore the hash function and point addition, because they are relatively light weight computations comparing with pairing, scalar and exponentiation operations.

Scheme	Encryption	Decryption	Key Publish
CL-PKE	3P+1S+1E	1P+1S	2 Points
V-IBE	**3P+1S+1E**	**1P+1S**	**2 Point**
B-IBE	1P+1S+1E	1P+1S	0 Point

In both schemes parties can save some computations (two pairing computation in the check procedure) by checking a party's key (nickname or public key) once and one pairing operation by precompute g before sending more than one message to the party.

One advantage of our scheme is that our scheme cooperates seamlessly with the original IBE system. In fact, the original IBE can be deemed as the V-IBE with (∞,∞) (the identity of the group $\mathbb{G}_1$) as nickname and q as PrKeyR for all entities. If the entity wants to use the "Nickname" system, it can use the original IBE system with some extension without modifying existing functions. All what it needs to do is to select a private key t and publish (tP, tP_{pub}) by itself. If one entity doesn't support the "Nickname" system in crypto-protocols, the entities can degenerate the security scheme to the basic IBE scheme gracefully. But the check procedure needs a minor modification to allow $N_1, N_2 \in \mathbb{G}_1$ instead of $\mathbb{G}_1^*$.

9 Conclusion

By introducing a new concept "Nickname", we modified Boneh-Franklin's IBE scheme by introducing a new pair of public and private keys to remove the key escrow function inherent in the IBE system. We find the new scheme inherits the basic property of the IBE system to enable part of the public key to be an arbitrary string, but at the same time removes the key escrow function without seriously increasing the PKC's burden. We also, using this variant, produce a signature scheme and an authenticated key agreement, and show one method to integrate our scheme into a hierarchial identity based public key encryption system.

References

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless Public Key Cryptography", Advances in Cryptology-Asiacrypt '2003, Lecture Notes in Computer Science, Springer-Verlag,to appear.

[2] D. Boneh and M. Franklin, "Identity Based Encryption from The Weil Pairing", extended abstract in Advances in Cryptology-Crypto 2001, Lecture Notes in Computer Science, Vol. 2139, Springer-Verlag, pp. 231-239, 2001.

[3] L. Chen, C. Kudla, "Identity Based Authenticated Key Agreement from Pairings", Cryptology ePrint Archive, Report 2002/184.

[4] C. Cocks, "An Identity Based Encryption Scheme Based on Quadratic Residues", Cryptography and Coding, LNCS 2260, pp. 360-363, 2001.

[5] W. Diffie and M.E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory 22, pp. 644-654,1976.

[6] P. Gemmel, "An Intoduction to Threshold Cryptography", CryptoBytes, a technical newsletter of RSA Laboratories, Vol. 2, No. 7, 1997.

[7] C. Gentry and A. Silverberg, "Hierarchical ID-Based Cryptography", Proceedings of Asiacrypt 2002, LNCS 2501 pp. 548-566, 2002.

[8] F. Hess, "Efcient Identity Based Signature Schemes Based on Pairings". In K. Nyberg and H. Heys, editors, Selected Areas in Cryptography 9th Annual International Workshop, SAC 2002, volume 2595 of LNCS, pages 310-324. Springer-Verlag, 2003.

[9] U. Maurer and S. Wolf, "Diffie-Hellman Oracles", Advances in Cryptology - CRYPTO '96 Proceedings, pp.268-282, Springer-Verlag, 1996.

[10] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", in Advances in Cryptology-Crypto '84, Lecture Notes in Computer Science, Vol. 196, Springer-Verlag, pp. 47-53, 1984.

[11] J. Silverman, "The Arithmetic of Elliptic Curve", Springer-Verlag, 1986.