REMOVE KEY ESCROW FROM THE IDENTITY-BASED ENCRYPTION SYSTEM

Zhaohui Cheng, Richard Comley and Luminita Vasiu
School of Computing Science, Middlesex University
White Hart Lane, London N17 8HR, United Kingdom
{m.z.cheng,r.comley,l.vasiu}@mdx. ac.uk

Abstract Key escrow is an inherent property in the current proposed Identity-Based Encryption (IBE) systems. However the key escrow is not always a good property for all applications. In this paper, we present a scheme which removes the key escrow from the IBE system proposed by Bonch and Franklin, while at the same time maintaining some important properties of the 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, Pairing

1 Introduction

Since the landmark paper "New directions in cryptography" [7] was published in 1976, public key systems have been playing a fundamental role in the modern information security society. To address the security threat of the "man-in-the-middle" attack, complicated public key certification systems have been developed for years. But the widespread deployment of public key systems depends heavily on the certification distribution systems which suffer from a scalability problem.

In an attempt to simplify the certification management in a Public Key Center (PKC), in 1984 Shamir [13] first formulated the concept of Identity-Based Cryptography (IBC) in which a public key is the identity (an arbitrary string) of an entity. Shamir presented an identity-based signature scheme in [13] and more signature schemes were proposed later. However constructing a practical Identity-Based Encryption (IBE) scheme has been an open problem for about twenty years. Recently Boneh and Franklin [3] and Cocks [5] presented two different systems separately. Boneh-Franklin's scheme has drawn much attention

because of its provable security and efficiency in practice. Our work is based on this scheme.

In an IBE system there are four algorithms: (1) **Setup** generates the 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 $ID \in \{0,1\}^*$ which is the identity of an entity, (3) **Encrypt** encrypts messages using the public key *ID,* and (4) **Decrypt** decrypts messages using the corresponding private key.

Because an entity's identity (ID) is used as the public key directly, some interesting usages of an IBE can be naturally introduced. For example an ID can include the public key expiry time, or differentiate the entity's credentials. On the other hand a special property is inherent in the proposed IBE scheme. In Shamir's scheme, the PKC uses the **Extract** algorithm to generate a private key corresponding to the public *ID.* Hence the PKC knows all the entities' private keys. This property is called *"key escrow".* Because the proposed scheme [3] and [5] follow Shamir's scheme to setup systems, they also inherit the key escrow function. However the key escrow function is not necessary for all types of applications and a cryptosystem with a key escrow property has some serious disadvantages. For example once the master-key is exposed, all the entities' private keys are leaked in principle and all the prior communication information is under threat of exposure. Some mechanisms can be used to increase the security of the master-key, for example the threshold cryptography [8]. Gentry and Silverberg presented a method in a hierarchical ID-based scheme [9] to restrict the key escrow function in small areas. But the existence of a master-key is still a threat to an entity's privacy. In [1] Al-Riyami and Paterson introduced the concept of "Certificateless Public Key Cryptography" (CL-PKC) and presented a scheme which removes the key escrow property successfully. In this paper, we introduce the *"nickname"* concept and present another variant of Boneh-Franklin's IBE system without the key escrow function.

The rest of this paper is structured as follows. In section 2, we describe the original Boneh-Franklin's IBE scheme which is the basis of our variant, and we also briefly introduce the bilinear map which is the basic mathematical tool used in the scheme. In the next section, we present our scheme to show how to remove the key escrow function. A security analysis of our variant is presented in section 4. Section 5 and 6 is a signature scheme and an authenticated key agreement based on our variant separately. We show how to integrate our scheme into a hierarchial identity-based public key encryption system in section 7. Finally we make a comparison with the CL-PKC scheme.

2 Boneh-Franklin's IBE Scheme

Boneh-Franklin's IBE scheme is the first efficient and security provable identity-based encryption scheme, which is based on a "bilinear map" (pairing) $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic groups of large prime order q. The bilinear map has the following properties:

1 Bilinear: For all $P, Q, R, S \in \mathbb{G}_1$, $\hat{e}(P+Q, R+S) = \hat{e}(P,R)\hat{e}(P,S)\hat{e}(Q,P)\hat{e}(Q,S)$.

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 $Q = 0_{\mathbb{G}_1}$. $0_{\mathbb{G}_1}$ and $1_{\mathbb{G}_2}$ are the identity of two groups respectively. In [3], the concrete IBE uses an admissible map with a distortion map to achieve the non-degeneracy.

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

The modified Weil and Tate pairings [14] on elliptic curves can be used to build such bilinear maps. The security of Boneh-Franklin's scheme is based on an assumption of the hardness of the "Bilinear Diffie-Hellman" (BDH) problem.

ASSUMPTION 1 **BDH Assumption.** *Let $\mathcal{G}$ be a BDH parameter generator with a security parameter 1^k. Define*

$$Adv_{\mathcal{G},\mathcal{A}}(k) = Pr[\mathcal{A}(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, aP, bP, cP) = \hat{e}(P,P)^{abc} \mid \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle \leftarrow \mathcal{G}(1^k), P \leftarrow \mathbb{G}_1, a, b, c \xleftarrow{R} \mathbb{Z}_q^*].$$

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

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

Setup: Given a security parameter 1^k, the parameter generator follows the steps.

1 generate two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ 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 integer $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$.

3 pick four cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*, H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n, H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$ and $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$ 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** = $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2, H_3, H_4 \rangle$. s is the **master-key** of the system.

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

Encrypt: Given a plaintext $m \in \mathcal{M}$, the ID of an entity and the public parameters **params,** follow the steps:

1 pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, m)$.

2 compute $Q_{ID} = H_1(ID)$ and $g = \hat{e}(P_{pub}, Q_{ID})$.

3 set the ciphertext to $C = \langle rP, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma) \rangle$.

Decrypt: Given a ciphertext $\langle U, V, W \rangle \in \mathcal{C}$, a private key d_{ID} and the system parameters **params,** perform the following steps.

1 compute $g' = \hat{e}(U, d_{ID})$ and $\sigma' = V \oplus H_2(g')$.

2 compute $m' = W \oplus H_4(\sigma')$ and $r' = H_3(\sigma', m')$

3 If $U \neq r'P$, reject the ciphertext, else return m' as the plaintext.

The consistency of the scheme follows from the bilinearity of $\hat{e}$. Boneh and Franklin proved that the scheme is semantically secure against the adaptive chosen ciphtertext attack (IND-CCA) [2] [3] in the random oracle model [4].

3 Our Variant of Boneh-Franklin's IBE system

Based on Boneh-Franklin's scheme, we introduce another public and private key pair $\langle N_{ID}, t \rangle$ into the scheme to remove the key escrow function. The private key t, a random integer in $\mathbb{Z}_q^*$, is only owned by the entity with an identity ID (we use entity ID to refer to the entity with the identity ID in the remaining part of the paper). In our scheme the encryption and decryption operations not only depend on the public key *ID* (in fact Q_{ID}) and the private key d_{ID}, but also on the second public key N_{ID} and the corresponding private key t. We name the public keys $\langle ID, N_{ID} \rangle$ as $\langle$*ID, Nickname*$\rangle$ and the private keys $\langle d_{ID}, t \rangle$ as $\langle$*PrKeyL, PrKeyR*$\rangle$. Because only entity ID knows *PrKeyR,* we can prove that the key escrow function in the PKC is removed. The effect of introducing $\langle N_{ID}, t \rangle$ is discussed after the description of the scheme's details. We can find that to publish a nickname is not a serious new burden for a PKC. For simplicity we name our system as **V-IBE** and

Boneh-Franklin's scheme as **B-IBE** in the following sections.

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

Setup: As the one in Boneh-Franklin's scheme.

Extract: Identical to **Extract** in Boneh-Franklin's scheme.

Publish: Given the system parameters **params,** an entity selects a random $t \in \mathbb{Z}_q^*$, and computes $N_{ID} = \langle N_1, N_2 \rangle = \langle tP, tP_{pub} \rangle$. The entity can ask the PKC to publish this extra parameter N_{ID} or publish it by itself or via any directory service as a nickname. Note that this publishing operation has no security requirement.

Encrypt: Given a plaintext $m \in \mathcal{M}$, the identity ID, public parameters **params** and the nickname $N_{ID} = \langle N_1, N_2 \rangle$ corresponding to ID, the following steps are performed.

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 pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, m)$.

3 compute $Q_{ID} = H_1(ID)$ and $g = \hat{e}(P_{pub} + N_1, Q_{ID})$.

4 set the ciphertext to $C = \langle rP, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma) \rangle$.

Decrypt: Given a ciphertext $\langle U, V, W \rangle \in \mathcal{C}$, d_{ID}, t and system parameters **params,** follow the steps:

1 compute $g' = \hat{e}(U, d_{ID} + tQ_{ID})$ and $\sigma' = V \oplus H_2(g')$.

2 compute $m' = W \oplus H_4(\sigma')$ and $r' = H_3(\sigma', m')$.

3 If $U \neq r'P$, reject the ciphertext, else return m' as the plaintext.

The consistency of the scheme can be verified by

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

Hence σ' in decryption equals σ in encryption. Thus, applying decryption on a ciphertext recovers the original message m.

Based on the BDH and another assumption stated in the next section, we can prove that the variant is secure against the adaptive chosen ciphertext attack (IND-CCA) in the random oracle model. Moreover this

scheme achieves some special properties that make it different from the normal public key systems and the existing identity-based encryption schemes.

CLAIM 1 **No more key escrow.** *Without knowing the private key t (PrKeyR) of an entity, an adversary cannot decrypt a message encrypted for the entity, even with the knowledge of the master-key s.*

This claim follows from Theorem 1 in the following section.

CLAIM 2 **Partially identity-based.** *Without knowing d_{ID} (PrKeyL) of an entity identified by the ID, an adversary cannot decrypt a message encrypted for the entity even if the adversary replaces the entity's nickname N_{ID} with its own choice.*

This claim follows from Theorem 2 in the following section. Because of this property, some special usages of the original IBE are still applicable in our scheme, e.g. an entity's ID appending with expiry time or credentials.

REMARK 1 **Loosely binding nicknames.** *The extra public key parameter N_{ID} introduced in our scheme need not be bound strictly (by secure method) to the entity ID. N_{ID} can be distributed through an unsafe channel as the entity's nickname. If Alice wants to send a message to Bob, but does not know Bob's nickname, she can ask Bob directly or query the PKC or any directory service publishing Bob's nickname. Because of Claim 2, the security of the communication cannot be compromised by Eve who launches the man-in-the-middle attack and changes Bob's nickname with her own choice except that Eve is the PKC. This characteristic differentiates our scheme from the normal certification-based public key systems. In [1], a simple way is presented to thwart the PKC to impersonate another entity in the man-in-the-middle attack. The basic idea is to bind entity A's identity ID_A and nickname N_A with A's real public key Q_A by re-defining $Q_A = H_1(ID_A \| N_A)$. If the PKC impersonates entity A, there will be two valid private keys for ID_A with different nicknames which can only be generated by the PKC.*

REMARK 2 **Forward security of the master key.** *Our scheme introduces an extra public and private key pair $\langle N_{ID}, t \rangle$ and only the entity ID knows the private key t. Hence even if the master key s of the PKC is leaked, the prior communications with destination to entity ID would not be exposed, but the following communication would become vulnerable to the man-in-the-middle attack.*

4 The V-IBE's Security

Before defining the security of the scheme, we elaborate two primitive foundations of the variant.

Firstly we prove that based on the BDH assumption, it is hard for the PKC to compute g' in decryption, even though it knows the master key s. To construct g', the PKC needs to use the available information $(s, P, U = rP, Q_{ID} = aP, N_{ID} = \langle tP, tP_{pub} \rangle)$ 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* $a, r, t \xleftarrow{R} \mathbb{Z}_q^*$ *and* s *is a fixed element in* $\mathbb{Z}_q^*$, *based on the 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 a subroutine to solve the BDH problem. Given a BDH challenge (P, aP, bP, cP), $\mathcal{B}$ randomly selects an element s from $\mathbb{Z}_q^*$ and passes (s, P, aP, bP, cP) as the challenge to $\mathcal{A}$. Upon receiving 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 does $\mathcal{B}$ because if $R = \hat{e}(P, P)^{ab(s+c)}$, $\mathcal{B}$'s response is $\hat{e}(P, P)^{ab(s+c)}\hat{e}(aP, bP)^{-s} = \hat{e}(P, P)^{abc}$.

Secondly we show that if an adversary without the master key wants to compute $g' = \hat{e}(rP, Q_{ID})^{(s+t)}$ in decryption, it needs to solve some hard problem. 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 as to compute $g' = \hat{e}(U, Q_{ID})^j$. But by applying the check step, the adversary needs to find $N_2 = tsP = (j - s)sP$ to pass the check step. If the adversary successfully finds N_2, then it is able to compute $s^2P = N_2 - jsP$. Given $(\mathbb{G}_1, q, P, sP)$ to compute s^2P is a squaring-DH problem in group $\mathbb{G}_1$, which is as hard as a normal DH problem because the order of $\mathbb{G}_1$ is known [12]. If an adversary $\mathcal{A}$ knows t and can compute g', we can slightly modify $\mathcal{A}$ to solve the BDH problem. Given a BDH problem (P, sP, aP, rP) where $s, a, r \xleftarrow{R} \mathbb{Z}_q^*$, after finding $N_1 = tP$, $\mathcal{A}$ computes $R = \hat{e}(P, P)^{sar}\hat{e}(tP, P)^{ra}$ but outputs $R \cdot \hat{e}(rP, aP)^{-t} = \hat{e}(P, P)^{sar}$. The output is just the solution to the BDH problem. Note that a legitimate party has t and saP to compute R. If $\mathcal{A}$ does not know t and $j = s + t \mod q$, it seems hard to find such N_1 and N_2 satisfying the check requirement and at the same time making the computation of g' easy. Based on this evaluation, we propose an assumption.

ASSUMPTION 2 *Given* $(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, sP, aP)$, *where* $s, a \xleftarrow{R} \mathbb{Z}_q^*$, *based on the BDH assumption, it is hard to find* $N_1, N_2 \in \mathbb{G}_1^*$ *satisfying* $\hat{e}(N_1, sP) = \hat{e}(N_2, P)$ *and at the same time making computation* $\hat{e}(P, P)^{sar}$. $\hat{e}(N_1, P)^{ra}$ *with* $rP \xleftarrow{R} \mathbb{G}_1$ *easy (here "easy" means existing a randomized polynomial time algorithm). (We refer to the assumption as a* **Bilinear EQuation (BEQ)** *assumption.)*

Now by defining two types of adversaries, which correspond to an adversary with and without the master-key respectively, we state the security analysis in the following two theorems.

Definition: Type-I Attack

An adversary with the *master-key* launches a Type-I attack by taking one or more of the following actions interacting with a challenger following from the IND-CCA notion.

1 Query the nickname of any entity ID_i.

2 Publish a nickname for any entity ID_i.

3 Extract *PrKeyL* of any entity ID_i. In fact because the adversary has the master-key, it can compute *PrKeyL* of any entity. But we still assume that the adversary issues Extract query to get the *PrKeyL* from the challenger for simplicity.

4 Extract *PrKeyR* of any entity ID_i but ID_{ch}. However querying *PrKeyR* of a nickname published by the adversary is prohibited because it is unreasonable to require that the challenger knows such value which implies that the challenger can solve the discrete logarithm problem.

5 Be challenged on the chosen ID_{ch} by providing two messages m_0, m_1. Note that the nickname N_{ch} of entity ID_{ch} is not the one published by the adversary. Hence it means that although the adversary can replace N_{ch} in some phase, it must be challenged on ID_{ch}'s original nickname. Following the IND-CCA notion, the challenger randomly chooses $b \in \{0, 1\}$ and provides the ciphertext of m_b.

6 Issue a decryption query $\langle ID_i, C_i \rangle$. The adversary is prohibited from making a decryption query on the challenge ciphertext for the combination of identity ID_{ch} and the original N_{ch}.

If the adversary with the master-key also changes the nickname N_{ch} of the entity ID_{ch} on which it wants to be challenged, it knows both d_{ch} and t_{ch}. Hence the scheme cannot protect the information encrypted under

ID_{ch} and the changed nickname. In traditional public key cryptosystems this attack is not prevented either. This is the reason for the rules in the challenge phase. In the IND-CCA model, an adversary can continue to ask queries after the challenge phase. The advantage of an adversary is defined as the amount by which the probability of guessing the correct b exceeds $\frac{1}{2}$ (i.e. Advantage=max {Pr[Guessing the correct b]-$\frac{1}{2}$,0}).

THEOREM 1 *If there exists a Type-I IND-CCA adversary $\mathcal{A}$ with non-negligible advantage ϵ against V-IBE, then there exists an adversary $\mathcal{B}$ which can solve the BDHP with non-negligible advantage in the random oracle model.*

Definition: Type-II Attack

An adversary without the *master-key* launching a Type-II attack can take one or more of the following actions when interacting with a challenger.

1 Query the nickname of any entity ID_i.

2 Publish a nickname for any entity ID_i.

3 Extract *PrKeyL* of any entity ID_i except ID_{ch}.

4 Extract *PrKeyR* of any entity ID_i. But the adversary should not query *PrKeyR* of a nickname published by itself.

5 Be challenged on the chosen ID_{ch} by providing two messages m_0, m_1. Note that there is no requirement on the nickname of ID_{ch}. Hence the adversary can be challenged on an entity whose nickname is published by the adversary. The challenger randomly chooses $b \in \{0, 1\}$ and provides the ciphertext of m_b.

6 Issue a decryption query $\langle ID_i, C_i \rangle$. The adversary is not allowed to query on the challenge ciphertext for the combination of identity ID_{ch} and the nickname used in the challenge query.

The adversary can query private *PrKeyL* of any entity ID_i except ID_{ch} and can publish a nickname for any entity. The advantage is defined similarly to the one for the Type-I adversary.

THEOREM 2 *If there exists an IND-CCA Type-II adversary $\mathcal{A}$ against V-IBE with advantage ϵ, then there exits an adversary $\mathcal{B}$ which can solve the BEQ problem with non-negligible advantage in the random oracle model.*

The proofs of the above two theorems are essentially similar to the proofs of Theorem 1 and 2 in the CL-PKC [1], but with different assumptions (the authors proposed a general BDH assumption in [1]).

5 A Signature Scheme Based on Our Variant

We describe a public key signature (PKS) scheme based on a provably secure signature scheme in [10] and our variant. The PKS scheme can be specified by algorithms: **Setup**, **Extract**, **Publish**, **Sign** and **Verify**.

Setup: Given a security parameter 1^k, the parameter generator follows the steps.

1 generate two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ 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 two cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \{0,1\}^* \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$.

The system parameters are **params**$= \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2 \rangle$. s is the **master-key** of the system.

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

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

Sign: To sign a message $m \in \mathcal{M}$ using the private key $\langle d_{ID}, t \rangle$ of entity ID, the following steps are performed.

1 choose an arbitrary point $P_1 \in \mathbb{G}_1^*$ and pick a random integer $k \in \mathbb{Z}_q^*$.

2 compute $r = \hat{e}(kP_1, P)$ and $v = H(m, r)$.

3 compute $Q_{ID} = H_1(ID)$ and $U = v(d_{ID} + tQ_{ID}) + kP_1$.

4 output as the signature $\langle U, v \rangle$.

Verify: To verify a signature $\langle U, v \rangle$ of entity ID with nickname $N_{ID} = \langle N_1, N_2 \rangle$ on a message $m \in \mathcal{M}$, 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 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 consistency of the scheme 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 protocol which extends Smart's protocol [15].

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

Upon the completion of message exchanges, A and B first check the exchanged nickname (N_{ID}^B and N_{ID}^A respectively). After that A computes $K_A = \hat{e}(Q_{ID}^B, P_{pub} + N_1^B)^x \cdot \hat{e}(d_{ID}^A + aQ_{ID}^A, yP)$, 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 the secret key $K = K_A = K_B$ is shared between A and B.

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

Although A and B can use $H(K \| xyP)$ as the shared key, where H is a proper hash function to achieve forward security, Shim's protocol and its descendant [6] are vulnerable to the man-in-the-middle attack launched by the PKC. The new variant still suffers from such attack if the PKC replaces the nicknames in the two messages with its own selections. However we can use the same method mentioned in Section 3 to thwart such attacks.

7 Hierarchical PKE

In [9] Gentry and Silverberg 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 the system, every entity is located in one level of a hierarchical system. Except the root entity, every entity is identified by an ID-tuple which identifies every ancestor along the path to the root. The major steps of our scheme are identical

to the ones in [9].

Root Setup: Given a security parameter 1^k, the parameter generator follows the steps.

1 generate two 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 $P_0 \in \mathbb{G}_1$.

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

3 pick two cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ 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 ID_t, select a random $b_t \in \mathbb{Z}_q^*$ and compute the nickname $N_t = \langle N_1^t, N_2^t \rangle = \langle b_t P_0, b_t Q_0 \rangle$.

Encryption: To encrypt $m \in \mathcal{M}$ with the ID-tuple $(ID_1, \ldots, ID_t)$ and the corresponding nicknames $N_i = \langle N_1^i, N_2^i \rangle$ for $1 \leq i \leq t$, take the following steps:

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

3 choose random $r \in \mathbb{Z}_q^*$, and compute cyphertext
$C = \langle U_0, U_2, \ldots, U_t, V \rangle = \langle rP_0, rP_2, \ldots, rP_t, m \oplus H_2(g^r) \rangle$,
where $g = \hat{e}(Q_0 + N_1^t, P_1) = \hat{e}(s_0 P_0, P_1) \cdot \hat{e}(b_t P_0, P_1)$.

Decryption: To decrypt the ciphertext $C = \langle U_0, U_2, \ldots, U_t, V \rangle \in \mathcal{C}_t$ for an entity in level t with the 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 The CL-PKC

In the above sections we have shown that all the cryptosystems supported by the CL-PKC can be realized using our variant. In fact, the public key in the CL-PKC is essentially the same as the nickname in our scheme. Hence, our variant is an alternative implementation of the CL-PKC but based on a different hardness assumption.

Our scheme is slightly slower than the CL-PKC, because our scheme needs an extra point addition operation. However the point addition is very fast compared to the pairing computation or the scalar operation. The following table compares the complexity of the two schemes and B-IBE (*P* for pairing computation, *S* for scalar operation and *E* for exponentiation). We ignore the hash operation and the point addition, because the numbers of hash operations in all schemes are equal and the point addition is a very lightweight computation compared to the 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 Points**
B-IBE	1P+1S+1E	1P+1S	0 Point

In both schemes (CL-PKE and V-IBE) entities can save two pairing computation in the check procedure by checking an intended entity's key (the nickname in V-IBE or the public key in CL-PKE) once and save one pairing operation by pre-computing g before sending more than one message to the intended entity.

A good property of our scheme is that it cooperates seamlessly with the original IBE system. In fact, the original IBE can be deemed as a V-IBE with $\langle \mathcal{O}, \mathcal{O} \rangle$ ($\mathcal{O}$ is the identity of group $\mathbb{G}_1$) as a nickname and q as *PrKeyR* for all entities. If an entity wants to use the "nickname" system, it can use the original IBE implementation by slightly modifying the existing functions to include the presented extension. After that, all that an entity needs to do is to select a private key t and publish $\langle tP, tP_{pub} \rangle$ by itself or via a directory service. If a peer entity does not support the nickname system in a crypto-protocol, the entities can degenerate the security scheme to the basic IBE scheme gracefully. To do this 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 modify Boneh-Franklin's IBE scheme to remove the inherent key escrow function. We find that 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 necessarily increasing the PKC's burden. Using this variant we extend a signature scheme and an authenticated key agreement to remove the key escrow property. We also 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, LNCS 2894, 2003.

[2] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations among notions of security for public-key encryption schemes", In Advances in Cryptology CRYPTO 98, LNCS 1462, 1998.

[3] D. Boneh and M. Franklin, "Identity Based Encryption from The Weil Pairing", extended abstract in Advances in Cryptology-Crypto 2001, LNCS 2139, 2001.

[4] M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Desiging Efficient Protocols", Proc. of First ACM Conference on Computer and Communication Security, November 1993.

[5] C. Cocks, "An Identity Based Encryption Scheme Based on Quadratic Residues", Cryptography and Coding, LNCS 2260, 2001.

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

[7] W. Diffie and M.E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory 22,1976.

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

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

[10] F. Heβ, "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, LNCS 2595, 2003.

[11] D. L. Long and A. Wigderson, "The discrete logarithm problem hides O(log n) bits", SIAM J. Computing, 17(2), April 1988.

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

[13] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", in Advances in Cryptology-Crypto '84, LNCS 196, 1984.

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

[15] N. P. Smart, "An Identity Based Authenticated Key Agreement Protocol Based on the Weil Pairing", Electronics Letters 38 (2002), pp. 630–632, 2002