Fully Secure Identity-Based Encryption Without Random Oracles: A variant of Boneh-Boyen HIBE

Yu Chen

School of Electronics Engineering and Computer Science
Peking University, Beijing, China
Email: cycosmic@gmail.com

Abstract. We present an Identity-Based Encryption (IBE) scheme that is fully secure without random oracles and has several advantages over previous such schemes - namely, computational efficiency, shorter public parameters, and simple assumption. The construction is remarkably simple and the security reduction is straightforward. We first give our CPA construction based on the decisional Bilinear Diffie-Hellman (BDH) problem, then archiving the CCA security by employing secure symmetric-key encryption algorithm. Additionally, we transform the CPA construction into a new signature scheme that is secure under the computational Diffie-Helleman assumption without random oracles.

Key words: identity based encryption, efficient, fully secure, standard model

1 Introduction

The concept of IBE was introduced by Shamir [1] in 1984. About twenty years later, Cocks [2], Boneh and Franklin [3] and Sakai et al. [4] presented three IBE solutions in 2001. Cock's scheme is based on the quadratic residuosity problem, which relies on the hardness of factoring. Both Boneh and Franklin scheme (BF-IBE) and Sakai et al. solution are based on groups with efficiently computable bilinear maps. The security of either scheme was only proven in the random oracle model.

Following the breakthough results in 2001, there has been significant progress in realizing IBE in the standard model. First, Canetti, Halevi, and Katz [5] suggested a weaker security notion for IBE, known as *Selective*-ID model, relative to which they were able to build an inefficient but secure IBE scheme without using random oracles. Subsquently, Boneh and Boyen [6] presented two very efficient IBE systems ("BB_1" and "BB_2") secure in the *Selective*-ID model, without random oracles. In Crypto 2004, the same authors [7] then proposed a coding-theoretic extenstion to their "BB_1" scheme which was proved to be fully secure for adaptive identity without random oracles. However, their construction is polynomial in all parameters, which make it impractical. It is mostly viewed as an existense proof of fully secure IBE builting with a polynomially bounded reduction from the underlying complexity assumption. In EuroCrypto 2005, Waters [8] greatly simplified the extenstion in [7] and substantially improved the efficiency. One drawback of Waters scheme [8] is that the public parameters consisted of $n + 4$ group elements which grows linear to the security parameter n. In the same year, Naccache [9] described a variant of Waters scheme which divided the system parameters by a factor l but at the cost of reducing the security by l bits. Independantly, Chatterjee and Sarkar [10] addressed the same issue and proposed a generalisation of Waters scheme, investigated how to find a trade-off between the smallness of parameters and the tightness of security reduction.

In Eurocrypt 2006, Gentry [11] proposed an IBE scheme with short public parameters. Although the Gnetry IBE scheme achieved security in the standard model, it did so at the cost of using a complicated assumption called the decisional q-ABDHE assumption. In addition to the added complexity, the actual assumption used in the proof is dependent on the number of private key queries the adversary makes. In Crypto 2009, Waters [12] presented a new methodology named *Dual System Encryption* which results in *fully* secure IBE and HIBE systems under simple assumption and with ciphertexts, private keys, and public parameters has constant size.

1.1 Our contribution

We present an IBE (HIBE) scheme that is fully secure without random oracles based on the simple and established *decisional* Bilinear Diffie-Hellman assumption. Our IBE scheme has ciphertexts, private keys, and public parameters each consisting of a constant number of group elements, and the security reduction is tighter than previous such schemes [8]. Additionally, like Waters, our CPA construction can also be easily converted to a signature scheme where the underlying assumption is the computational Diffie-Helleman problem by applying Naor's technique.

Another contribution of this paper is our proof technique. We embed an unbalanced linear combination structure into the generation of public key. This technique enable us to partition the whole identity space into two orthogonal subspaces and therefore achieve the full security without random oracles. The intuition of proof technique will be detailed in Section 4.3.

1.2 Organization

We organize the rest of the paper as follows. In Section 2 we give our security definitions. In the Section 3 we describe some necessary complexity assumptions. In Section 4 and Section 5 we present the CPA construction and CCA construction, respectively. In Section 6 we show how to transform the CPA construction to a signature scheme. Finally, we conclude in Section 7.

2 Security Definitions

2.1 IBE and HIBE

Following [13] [14], a Hierarchical Identity Based Encryption (HIBE) systems consists of four algorithms: Setup, KeyGen, $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$. In HIBE, identities are vectors; a vector of dimension k represents an identity at depth k. The Setup algorithm generates system parameters, denoted by *params*, and a master secret *master-key*. Note that, for a HIBE of height ℓ (henceforth denoted as ℓ-HIBE) any identity ID is a tuple $(\mathrm{I}_1, \ldots, \mathrm{I}_k)$ where $1 \leq k \leq \ell$.

Setup. Takes a security parameter κ and returns *params* and *master-key*. Intuitively, the system parameters *params* will be publicly known, while the *master-key* will be known only to the Private Key Generator (PKG). We refer to the *master-key* as the private key at depth 0 and note that an IBE system is a HIBE where all identities are at depth 1. $\mathcal{M}$ is the message space, $\mathcal{C}$ is the ciphertext space.

KeyGen. Takes as input an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k)$ at depth k and the private key $d_{\mathsf{ID}|(k-1)}$ of the parent identity $\mathsf{ID}_{|k-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{k-1})$ at depth $k-1$, and then outputs a private key d_{ID} for identity ID.

Encrypt. Takes as input *params*, ID, and $M \in \mathcal{M}$. It returns a ciphertext $C \in \mathcal{C}$.

Decrypt. Takes as input *params*, $C \in \mathcal{C}$, and a private key d_{ID}. It returns $M \in \mathcal{M}$.

Boneh and Franklin [3] [15] define chosen ciphertext security and semantic security for IBE (HIBE) schemes under a chosen identity attack using the following game between an adversary $\mathcal{A}$ and a challenger:

Setup. The challenger runs Setup algorithm and gives $\mathcal{A}$ the resulting system parameters *params*, keeping the *master-key* to itself.

Phase 1. The adversary $\mathcal{A}$ issues queries $q_1, \ldots, q_m$ where q_i is one of:

- Private key query $\langle \mathsf{ID}_i \rangle$. The challenger responds by running algorithm KeyGen to generate the private key d_i corresponding to the public key ID_i and sends d_i to the adversary.
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. The challeger responds by running algorithm KeyGen to generate the private key d_i corresponding to ID_i. It then runs algorithm $\mathsf{Decrypt}$ to decrypt the ciphertext C_i using the private key d_i and sends the resulting plaintext to the $\mathcal{A}$.

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

Challenge. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs an identity ID^* and two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ on which it wishes to be challenged. The only restriction is that $\mathcal{A}$ did not previously issue a private key query for ID^* or a prefix of ID^*. The challeger picks a random bit $\beta \in \{0, 1\}$ and sets the challenge ciphertext to $C = \mathsf{Encrypt}(params, M_\beta, \mathsf{ID}^*)$, which is sent to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ issues additional queries $q_{m+1}, \ldots, q_n$ where q_i is one of:

- Private key query $\langle \mathsf{ID}_i \rangle$ where $\mathsf{ID}_i \neq \mathsf{ID}^*$ and ID_i is not a prefix of ID^*.
- Decryption query $\langle C_i \rangle \neq \langle C \rangle$ for ID^* or any prefix of ID^*.

In both cases, the challeger responds as in Phase 1. These queries may be adaptive.

Guess. $\mathcal{A}$ eventually outputs a bit β' and wins if $\beta' = \beta$.

We refer to such an adversary $\mathcal{A}$ as an $\mathsf{IND\text{-}ID\text{-}CCA}$ adversary. $\mathcal{A}$'s advantage is defined as $Adv_{\mathcal{A}}(k) = |\Pr[\beta' = \beta] - 1/2|$. The probability is over the random bits used by the challenger and the adversary.

Definition 2.1 *An IBE or HIBE scheme $\mathcal{E}$ is said to be (t, q_E, q_D, ϵ)-adaptive identity, chosen ciphertext secure of for any t-time* $\mathsf{IND\text{-}ID\text{-}CCA}$ *adversary that makes at* *most q_E chosen private key queires and at most q_D chosen decryption queries we have $Adv_{\mathcal{A}} \leq \epsilon$. As shorthand, we say that $\mathcal{E}$ is (t, q_E, q_D, ϵ)* IND-ID-CCA *secure.*

Semantic Security. We define adaptive identity, chosen plaintext security for an IBE or HIBE scheme as in the preceding game, except that the adversary is not allowed to issue any decryption queries. The adversary can still make adaptive private key extraction queries.

Definition 2.2 *An IBE or HIBE scheme $\mathcal{E}$ is said to be (t, q_E, ϵ)-adaptive identity, chosen plaintext secure of for any t-time* IND-ID-CPA *adversary that makes at most q_E chosen private key queires we have $Adv_{\mathcal{A}} \leq \epsilon$. As shorthand, we say that $\mathcal{E}$ is (t, q_E, ϵ)* IND-ID-CPA *secure.*

2.2 Symmetric-Key Encryption

A symmetric-key encryption scheme SE consists of two algorithms $(\mathsf{Enc}, \mathsf{Dec})$. For a symmetric key sk, the encryption algorithm Enc encrypts a plaintext M as $C \leftarrow \mathsf{Enc}(sk, M)$; The decryption algorithm Dec decrypts a ciphertext C as $M = \mathsf{Dec}(sk, C)$. Moreover, we say that SE is length preserving if $|\mathsf{Enc}(sk, M)| = |M|$.

Definition 2.3 *A symmetric-key encryption scheme is secure in the* IND-CCA *sense if no PPT adversary $\mathcal{A}$ has a non negligible advantage in the following game.*

Setup. The challenger randomly chooses a symmetric key sk.

Phase 1. $\mathcal{A}$ starts probing the scheme by querying the encryption oracle $\mathsf{Enc}(sk, \cdot)$ and the decryption oracle $\mathsf{Dec}(sk, \cdot)$.

Challenge. In the challenge phase, $\mathcal{A}$ outputs two equal length messages (M_0, M_1) that were not submitted to $\mathsf{Enc}(sk, \cdot)$ or obtained from $\mathsf{Dec}(sk, \cdot)$ and gets $C = \mathsf{Enc}(sk, M_\beta)$ for a random bit $\beta \in \{0, 1\}$.

Phase 2. $\mathcal{A}$ issues new queries as in Phase 1 but is disallowed to ask for the decryption of C and the encryptions of M_0 and M_1.

Guess. $\mathcal{A}$ eventually outputs a guess β' for β.

$\mathcal{A}$'s advantage is defined by $Adv_{\mathcal{A}}(k) = |\Pr[\beta' = \beta] - 1/2|$.

We will use a length preserving IND-CCA secure symmetric-key encryption scheme in our construction. Such a scheme can be built by applying CMC [16] or EME [17] mode of operation to a block cipher, if the underlying block cipher is modeled as strong pseudorandom permutation (for example, AES [18] can be used).

3 Complexity Assumptions

We briefly review the facts about groups with efficiently computable bilinear map. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of large prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be the bilinear map between these two groups. A bilinear map satisfying the following three properties is said to be an admissible bilinear map.

1. Bilinearity. The map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is bilinear if $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$.
2. Non-degeneracy. The map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$.
3. Computability. There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}_1$.

Bilinear Map Parameter Generator. We say that a randomised algorithm $\mathcal{G}$ is a Bilinear Map parameter generator if (1) $\mathcal{G}$ takes a security parameter $k \in \mathbb{Z}^+$, (2) $\mathcal{G}$ runs in polynomial time in k, and (3) $\mathcal{G}$ outputs a k bits prime number p, the description of two groups $\mathbb{G}_1$, $\mathbb{G}_2$ of order p, and the description of an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. We denote the output of $\mathcal{G}$ by $\mathcal{G}(1^k) = \langle \mathbb{G}_1, \mathbb{G}_2, p, e\rangle$.

3.1 Decisional Bilinear Diffie-Hellman (BDH) Assumption

Suppose $\mathcal{G}(1^k) = \langle \mathbb{G}_1, \mathbb{G}_2, p, e\rangle$, the challenger chooses $a, b, c, z \in \mathbb{Z}_p$ at random and then flips a fair coin β. If $\beta = 1$ it outputs the tuple $(P, P_1 = aP, P_2 = bP, P_3 = cP, Z = e(P, P)^{abc})$. Otherwise, the challenger outputs the tuple $(P, P_1 = aP, P_2 = bP, P_3 = cP, Z = e(P, P)^z)$. Finally, the adversary outputs a guess β' for β. An adversary $\mathcal{B}$ has at least an ϵ advantage in solving the decisional BDH problem if

$$\left|\Pr\left[\mathcal{B}\left(P, P_1, P_2, P_3, e(P, P)^{abc}\right) = 1\right] - \Pr\left[\mathcal{B}\left(P, P_1, P_2, P_3, e(P, P)^z\right) = 1\right]\right| \geq \epsilon$$

where the probability is over the randomly chosen a, b, c, z and the random bits consumed by $\mathcal{B}$. We refer to the distribution on the left as $\mathcal{P}_{BDH}$ and the distribution on the right as $\mathcal{R}_{BDH}$.

Definition 3.1 *The decisional (t, ϵ)-BDH assumption holds if no t-time adversary has at least ϵ advantage winning the above game.*

3.2 Computational Diffie-Hellman (DH) Assumption

The challenger chooses $a, b \in \mathbb{Z}_p$ at random and outputs $(P, P_1 = aP, P_2 = bP)$. The adversary then attempts to output abP. An adversary $\mathcal{B}$ has at least an ϵ advantage if

$$\Pr[\mathcal{B}(P, P_1, P_2) = abP] \geq \epsilon$$

where the probability is over the random a, b and the random bits consumed by $\mathcal{B}$.

Definition 3.2 *The computational (t, ϵ)-DH assumption holds if no t-time adversary has at least ϵ advantage winning the above game.*

4 Efficient IBE and HIBE Based on decisional BDH Without Random Oracles

We construct an efficient HIBE scheme that is fully secure without random oracles based on the decisional BDH assumption. In particular, this implies an efficient fully secure, chosen ciphertext secure IBE based on decisional BDH without random oracles. Our construction can be viewed as a variant of "BB_1" in [6]. We first present our scheme then describe its relation to the "BB_1".

4.1 CPA Construction

For now, we assume identities (ID_s)of depth ℓ are vectors of elements in $\{0,1\}^*$. We write $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_\ell)$. The j-th component corresponding to the identity at level j can be any arbitrary bitstring over $\{0,1\}^*$. The HIBE system works as follows:

Setup(ℓ): Run $\mathcal{G}$ on security parameter κ to generate $\langle \mathbb{G}, \mathbb{G}_1, p, e\rangle$. To generate system parameters for an HIBE of maximum depth ℓ, select a random generator P in $\mathbb{G}^*$, a random $a \in \mathbb{Z}_p^*$, and set $P_1 = aP$. Next, pick ℓ random elements $U_1, \ldots, U_\ell \in \mathbb{G}^*$ and a random element $P_2 \in \mathbb{G}^*$. Additionally, choose two collision resistant hash function $H_1 : \{0,1\}^* \to \mathbb{Z}_p$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_m$. (The choice of m will be determined later.) Let $\cdot$ denote the group operation in $\mathbb{G}_1$. The public parameters *params* and the master secret *master-key* are given by

$$params = (P, P_1, P_2, U_1, \ldots, U_\ell, H_1, H_2), \quad master\text{-}key = aP_2$$

For $j = 1, \ldots, \ell$, $\mathrm{I}_j \in \{0,1\}^*$, we define $F_j : \{0,1\}^* \to \mathbb{G}$ to be the function:

$$F_j(\mathrm{I}_j) = U_j + H_1(\mathrm{I}_j)P + H_2(\mathrm{I}_j)P_2 = U_j + w_jP + l_jP_2$$

KeyGen$(d_{\mathsf{ID}|j-1}, \mathsf{ID})$: To generate the private key d_{ID} for an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$ of depth $j \leq \ell$, pick random $r_1, \ldots, r_j \in \mathbb{Z}_p$ and output $d_{\mathsf{ID}} = (d_0, d_1, \ldots, d_j)$, where

$$d_0 = aP_2 + \sum_{i=1}^{j} r_iF_i(\mathrm{I}_i), d_i = r_iP \text{ for } 1 \leq i \leq j$$

Note that the private key for ID can be generated by an entity which processes a private key for $\mathsf{ID}_{|j-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{j-1})$. Indeed, let $d_{\mathsf{ID}|j-1} = (d'_0, \ldots, d'_{j-1})$ be the private key for $\mathsf{ID}_{|j-1}$. To generate d_{ID} pick a random $r_j \in \mathbb{Z}_p$ and computes $d_{\mathsf{ID}} = (d_0, d_1, \ldots, d_j)$ as follows.

$$d_0 = d'_0 + r_jF_j(\mathrm{I}_j); d_i = d'_i \text{ for } 1 \leq i \leq j-1; \text{ and } d_j = r_jP$$

In fact, any prefix of ID as well as the PKG can generate a private key d_{ID} for ID.

Encrypt$(params, \mathsf{ID}, M)$: To encrypt a message $M \in \mathbb{G}_1$ under the identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$, pick a random $c \in \mathbb{Z}_p$ and output

$$C = (A, B, C_1, \ldots, C_j) = (e(P_1, P_2)^c \cdot M, cP, cF_1(\mathrm{I}_1), \ldots, cF_j(\mathrm{I}_j))$$

Note that $e(P_1, P_2)$ can be precomputed once and for all so that encryption does not require any pairing computations. Alternatively, $e(P_1, P_2)$ can be included in the system parameters.

Decrypt(d_{ID}, C): Consider an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$. To decrypt a given ciphertext $C = (A, B, C_1, \ldots, C_j)$ using the private key $d_{\mathsf{ID}} = (d_0, d_1, \ldots, d_j)$, output

$$M = A \cdot \frac{\prod_{k=1}^{j} e(C_k, d_k)}{e(B, d_0)}$$

Indeed, for a valid ciphertext, we have

$$\frac{e(B, d_0)}{\prod_{k=1}^{j} e(C_k, d_k)} = \frac{e(P, P_2)^{ca} \prod_{k=1}^{j} e(P, F_k(\mathrm{I}_k))^{cr_k}}{\prod_{k=1}^{j} e(F_k(\mathrm{I}_k), P)^{cr_k}} = e(P_1, P_2)^c$$

4.2 Security

The HIBE scheme above is reminiscent of Boneh-Boyen HIBE (BB_1) [6] which is only known to be secure in *Selective*-ID model. Surprisingly, our choice of functions $F_1, \ldots, F_\ell$ enables us to prove our scheme fully secure without random oracles. We prove security of our HIBE scheme under decisional BDH assumption in groups generated by $\mathcal{G}$.

Theorem 4.1 *Our HIBE system is* IND-ID-CPA *assuming the decisional BDH problem is hard in groups generated by $\mathcal{G}$. Concretely, suppose there is an* IND-ID-CPA *adversary $\mathcal{A}$ that has advantage ϵ against the scheme. If $\mathcal{A}$ makes at most $q_E > 0$ private key extraction queries. Then there is an algorithm $\mathcal{B}$ that solves the decisional BDH generated by $\mathcal{G}$ with advantage at least:*

$$Adv_{\mathcal{B}} \geq \frac{\epsilon}{e \cdot q_E^\ell}$$

Proof. Suppose $\mathcal{A}$ has advantage ϵ in attacking the HIBE system. We show how to build an adversary $\mathcal{B}$ that uses adversary $\mathcal{A}$ against decisional BDH problem. Algorithm $\mathcal{B}$ is given as input a random 5-tuple (P, aP, bP, cP, Z) that either sampled from $\mathcal{P}_{BDH}$ (where $Z = e(P,P)^{abc}$) or from $\mathcal{R}_{BDH}$ (where Z is uniform and independent in $\mathbb{G}_1$). Algorithm $\mathcal{B}$'s goal is to output 1 if $Z = e(P,P)^{abc}$ and 0 otherwise. Set $P_1 = aP$, $P_2 = bP$, $P_3 = cP$. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an IND-ID-CPA game as follows.

Setup. To generate the system parameters, $\mathcal{B}$ randomly picks $s_i \in \mathbb{Z}_p$ and $t_i \in \mathbb{Z}_m$ at random for $i = 1, \ldots, \ell$, and assigns $U_i = s_iP - t_iP_2$. (s_i and t_i are kept internal to $\mathcal{B}$.) It gives $\mathcal{A}$ the system parameters $params = (P, P_1, P_2, U_1, \ldots, U_\ell)$. Note that the corresponding *master-key*, which is unknown to $\mathcal{B}$, is $aP_2 = abP \in \mathbb{G}$. From the perspective of the adversary $\mathcal{A}$ the distribution of the public parameters $(P, P_1, P_2, U_1, \ldots, U_\ell)$ are identical to the real construction.

Phase 1 - Private key queries. $\mathcal{A}$ issues up to q_E private key queries. Consider a query for the private key corresponding to $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_u)$ where $u \leq \ell$. To respond to this query, $\mathcal{B}$ first computes $H_1(\mathrm{I}_i) = w_i$ and $H_2(\mathrm{I}_i) = l_i$ for $1 \leq i \leq u$, then compute F_i functions taking I_i as inputs:

$$F_i(\mathrm{I}_i) = U_i + H_1(\mathrm{I}_i)P + H_2(\mathrm{I}_i)P_2 = s_iP - t_iP_2 + w_iP + l_iP_2$$

If $l_i = t_i$ the simulator aborts and randomly choose its guess β' of the challenge's value β. Otherwise, the simulator $\mathcal{B}$ picks $r_i, \ldots, r_u \in \mathbb{Z}_p$ at random and constructs the private key $d = (d_0, d_1, \ldots, d_u)$ as follows:

$$d_0 = \sum_{i=1}^{u} \left(r_i(s_i + w_i)P + r_i(l_i - t_i)P_2 - \frac{s_i + w_i}{u(l_i - t_i)} P_1 \right)$$

$$d_i = r_iP - \frac{1}{u(l_i - t_i)} P_1 (\text{ for } 1 \leq i \leq u)$$

Observe that

$$\begin{aligned}
d_0 &= \sum_{i=1}^{u} \left(r_i(s_i + w_i)P + r_i(l_i - t_i)P_2 - \frac{s_i + w_i}{u(l_i - t_i)} P_1 \right) \\
&= aP_2 + \sum_{i=1}^{u} \left(r_i(s_i + w_i)P + r_i(l_i - t_i)P_2 - \frac{s_i + w_i}{u(l_i - t_i)} aP - \frac{a}{u} P_2 \right) \\
&= aP_2 + \sum_{i=1}^{u} \left(r_i - \frac{a}{u(l_i - t_i)} \right) ((s_i + w_i)P + (l_i - t_i)P_2) \\
&= aP_2 + \sum_{i=1}^{u} \left(r_i - \frac{a}{u(l_i - t_i)} \right) (U_i + w_iP + l_iP_2) \\
&= aP_2 + \sum_{i=1}^{u} \tilde{r}_i F_i(\mathrm{I}_i) \\
d_i &= r_iP - \frac{1}{u(l_i - t_i)} P_1 = \left(r_i - \frac{a}{u(l_i - t_i)} \right) P = \tilde{r}_i P
\end{aligned}$$

We conclude that $d = (d_0, d_1, \ldots, d_u)$ is a valid private key for ID with the underlying random number $\tilde{r}_i = r_i - \frac{a}{u(l_i - t_i)}$. The simulator $\mathcal{B}$ is always able to perform this kind of construction iff $(l_i - t_i) \neq 0$ for all $1 \leq i \leq u$.

Challenge. When $\mathcal{A}$ decides that Phase 1 is over, it outputs two messages $M_0, M_1 \in \mathbb{G}_1$ and a target identity $\mathsf{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$ $(1 \leq k \leq \ell)$ on which it wishes to be challenged. The only constraint is that ID^* is not a prefix of any identity has been asked for private key in Phase 1. $\mathcal{B}$ first computes $H_1(\mathrm{I}_i^*) = w_i^*$ and $H_2(\mathrm{I}_i^*) = l_i^*$ for $1 \leq i \leq k$. If $l_i^* \neq t_i$ the simulator aborts and submits a random guess for β'. Otherwise, $\mathcal{B}$ computes $F_i^*(\mathrm{I}_i^*) = U_i + w_i^*P + l_i^*P_2 = (s_i + w_i^*)P$, then flips a fair coin β and responds the ciphertext

$$C = (A, B, C_1, \ldots, C_k)$$

where $A = Z \cdot M_\beta$, $B = P_3 = cP$, and $C_i = (s_i + w_i^*)P_3$. Note that $C_i = (s_i + w_i^*)P_3 = c(s_iP - t_iP_2 + w_i^*P + k_i^*P_2) = cF_i^*(\mathrm{I}_i^*)$, so we claimed that $(A, B, C_1, \ldots, C_l)$ is a valid ciphertext when $Z = e(P_1, P_2)^c = e(P, P)^{abc}$. It is easy to see that C is a valid encryption of M_β under ID^*. Otherwise, when Z is uniform and independent in $\mathbb{G}_1$ (when the input 5-tuple is sampled from $\mathcal{R}_{BDH}$) then C is independent of the simulator's choice β in adversary's view.

Phase 2 - Private key queries. $\mathcal{A}$ continues to issue queries. $\mathcal{B}$ responds as Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess $\beta' \in \{0, 1\}$. $\mathcal{B}$ ends its own game by outputting a guess as follows. If $\beta = \beta'$ then $\mathcal{B}$ outputs 1 meaning $Z = e(P, P)^{abc}$. Otherwise, it outputs 0 meaning $Z \neq e(P, P)^{abc}$.

When the input 5-tuple is sampled from $\mathcal{P}_{BDH}$ (where $Z = e(P, P)^{abc}$) then $\mathcal{A}$'s view is identical to its view in a real attack and therefore we have $|\Pr[\beta = \beta'] - 1/2| \geq \epsilon$. On the other hand, when the input 5-tuple is sampled from $\mathcal{R}_{BDH}$ (where Z is uniform in $\mathbb{G}_1$) then $\Pr[\beta = \beta'] = 1/2$. Therefore, with P, U_i uniform in $\mathbb{G}^*$, a, b, c uniform in $\mathbb{Z}_p$, Z uniform in $\mathbb{G}_1$, we have that

$$\begin{aligned}&\left|\Pr[\mathcal{B}(P, P_1, P_2, P_3, e(P,P)^{abc}) = 1] - \Pr[\mathcal{B}(P, P_1, P_2, P_3, Z) = 1]\right| \\ &\qquad \geq \left|\left(\frac{1}{2} \pm \epsilon\right) - \frac{1}{2}\right| = \epsilon\end{aligned}$$

To complete the proof of Theorem 4.1 it remains to calculate the probability that $\mathcal{B}$ aborts during the simulation. $\mathcal{B}$ may aborts simulation due to the two following events.

1. Event $\mathcal{E}_1$: $\bigvee_{i=1}^{u} l_i = t_i$ for any query in Phase 1 and Phase 2 when answering the private key extraction queries.
2. Event $\mathcal{E}_2$: $\bigvee_{i=1}^{k} l_i^* \neq t_i$ during the challenge phase.

$H_2(\cdot)$ is a collision resistant hash function, its outputs uniformly distribute in $\mathbb{Z}_m$, thus the probability of $\Pr[l_i = t_i]$ is $1/m$. For any private key extraction query related to a depth u identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_u)$ $(1 \leq u \leq \ell)$, the probability that $\mathcal{B}$ can generate the corresponding private key is

$$\Pr\left[\bigwedge_{i=1}^{u} l_i \neq t_i\right] = \left(1 - \frac{1}{m}\right)^{u} \geq \left(1 - \frac{1}{m}\right)^{\ell}$$

Suppose the maximum number of private key queries is q_E, then we have

$$\Pr[\neg\mathcal{E}_1] \geq \left(1 - \frac{1}{m}\right)^{\ell q_E}$$

According to the definition of event $\mathcal{E}_2$, it is easy to see that

$$\Pr[\neg\mathcal{E}_2 | \neg\mathcal{E}_1] = \left(\frac{1}{m}\right)^{k} \geq \left(\frac{1}{m}\right)^{\ell}$$

Therefore

$$\Pr[\mathcal{B} \text{ does not aborts}] = \Pr[\neg\mathcal{E}_2 | \neg\mathcal{E}_1]\Pr[\neg\mathcal{E}_1] \geq \left(\frac{1}{m}\right)^{\ell}\left(1 - \frac{1}{m}\right)^{\ell q_E}$$

A common estimate used here is $q_E = 2^{30}$ (suggested by Bellare and Rogaway [19]). We can optimize the probability by setting $m_{opt} = 1 + q_E$ (as we did in the simulation). Using m_{opt}, we have

$$\Pr[\mathcal{B} \text{ does not aborts}] = \frac{1}{q_E^{\ell}}\left(1 - \frac{1}{1 + q_E}\right)^{(1+q_E)\cdot\ell} \approx \frac{1}{e \cdot q_E^{\ell}}$$

If the adversary makes less queries the probability of not aborting can only be greater. This shows that $\mathcal{B}$'s advantage is at least $\epsilon/(e \cdot q_E^{\ell})$ as required. □

Remark 1. At first glance, there is an apparent paradox in the proof since it seems that the simulator algorithm could simply answer the challenge ciphertext itself by creating a private key for ID^*. In fact, for an arbitrary identity, the simulator cannot generate a private key and a valid ciphertext simultaneously. If the simulator can create a valid ciphertext for the challenge identity, which means $\bigvee_i^k l_i^* = t_i$). Then it cannot extract the corresponding private key.

Remark 2. Note that it is impossible for an adversary to generate a valid private key as the simulator does in simulation, because the structure of U_i is unknown to the adversary. One has a chance to generate a valid private key only when it can express U_i in an explicit form $s_i P + w_i P_2$, where $s_i \in \mathbb{Z}_p$, $w_i \in \mathbb{Z}_m$. This again is a hard problem.

4.3 Proof Technique

The proof technique we use is actually the "partitioning strategy" summarized by Waters in [12]. We reduce the security of our scheme to the underlying complexity assumption by building an algorithm $\mathcal{B}$ that partitions the identity space V into two subspaces: 1) V_1: identities of which it can create private keys; and 2) V_2: identities that it can use in the challenge phase. (It can embed the underlying complexity assumption instance into a valid ciphertext for the challenge identity.) We remark that the two subspaces are orthogonal, i.e. $V_1 \perp V_2$, $V = V_1 \oplus V_2$.

In order to achieve tight partition, we expect V_2 to be larger. This inspires us to embed an unbalanced structure to the public parameters and public key. Remember that F_j function can be viewed as a map to point function which maps an arbitrary identity to its underlying public key in $\mathbb{G}$:

$$F_j(\mathrm{I}_j) = U_j + Q_j, \text{ where } Q_j = H_1(\mathrm{I}_j)P + H_2(\mathrm{I}_j)P_2$$

The public key consists of two parts, public parameter U_j and the element Q_j of I_j. Both U_j and Q_j have the same structure, the linear combination based on 2-tuple generators (P, P_2): $x_1 P \pm x_2 P_2$, where $x_1 \in \mathbb{Z}_p$, $x_2 \in \mathbb{Z}_m$. Note that compared to first coefficient x_1, the second coefficient x_2 is chosen from a smaller space, so we refer to this combination as unbalanced structure. The partitioning is thus determined by implicitly assigning the unbalanced structure into public parameters U_j and explicitly embedding the unbalanced structure into Q_j.

As to implicitly assigning the unbalanced structure to U_j when the simulator run the Setup algorithm:

- Choosing the first coefficient s_j randomly from a large set $\mathbb{Z}_p$ enables U_j to be uniformly distributed in $\mathbb{G}$. If we drop the first term, then U_j does not uniformly distribute in $\mathbb{G}$ anymore.
- Choosing the second coefficient t_j from a small set $\mathbb{Z}_m$ enables the security reduction to be valid. If we drop the second term, alternatively means the second coefficient is 0. Then the adversary can always make the reduction invalid, i.e, issuing the private key query of which $H_2(\mathrm{I}_i) = 0$ and submitting the challenge identity of which $H_2(\mathrm{I}_i^*) \neq 0$.

As to explicitly embedding the unbalanced structure into Q_j

- Choosing the first coefficient w_j $(H_1(\mathrm{I}_j))$ randomly from a large set $\mathbb{Z}_p$ enables every identity could be well encoded into $\mathbb{G}$. (Introducing the first term makes every identity to be mapped to an unique element in $\mathbb{G}$. If we drop the first term, then the whole identity space will be mapped into a small set. In this case, it is easy for an adversary to find a collision.)
- Choosing the second coefficient l_j $(H_2(\mathrm{I}_j))$ from a small set $\mathbb{Z}_m$ enables the security reduction to be tight reduced to the underlying problem.

4.4 Comparison to Boneh-Boyen HIBE

Our scheme is quite similar to Boneh-Boyen HIBE [6]. The differences lie at the constructions of F_i functions. Let I_j be an identity of depth j, $H_1 : \{0,1\}^* \to \mathbb{Z}_p$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_m$ are two collision free hash functions. For $j = 1, \ldots, \ell$, the function F_j in Boneh-Boyen HIBE [6] is defined as

$$F_j(\mathrm{I}_j) = U_j + H_1(\mathrm{I}_j)P_1$$

whereas in our scheme, F_j is defined as

$$F_j(\mathrm{I}_j) = U_j + H_1(\mathrm{I}_j)P + H_2(\mathrm{I}_j)P_2$$

Remarkably, this small modification to F_j function enable us to achieve adaptive identity security (fully secure). The motivation of our design is easy to be understood from the above security proof.

We provide in Table 1 a comparison between our CPA construction and Waters scheme [8]. Let P be a pairing operation, G_i, E_i and I_i be a group operation, a group exponentiation and one group inversion in $\mathbb{G}_i$, respectively. Let k_i be the size of an element in $\mathbb{G}_i$. Reduction cost refers to the multiplicative ratio between the advantage of the adversary attacking the IBE scheme and the algorithm solving the underlying problem.

	Waters [8]	Our scheme
Public parameter size	$(n+4)k_1$	$4k_1$
Ciphertext size	$2k_1 + k_2$	$2k_1 + k_2$
Encryption	$\left(\frac{n}{2}+1\right)\mathrm{G}_1 + 1\mathrm{G}_2 + 2\mathrm{E}_1 + 1\mathrm{E}_2$	$2\mathrm{G}_1 + 1\mathrm{G}_2 + 2\mathrm{E}_1 + 1\mathrm{E}_2$
Decryption	$2\mathrm{P} + 1\mathrm{G}_2 + 1\mathrm{I}_2$	$2\mathrm{P} + 1\mathrm{G}_2 + 1\mathrm{I}_2$
Reduction factor	$\frac{1}{32(n+1)q_E}$	$\frac{1}{eq_E}$

[1] For security concern, n in [8] is suggested to be at least 128.

Table 1. Comparison with Waters scheme

5 CCA security

One way to achieve CCA security for our scheme is to follow the strategy suggested in [8]. Results of Canetti et al. [20], further improved upon by Boneh and Katz [21] show how to build a CCA secure ℓ-HIBE scheme from a CPA secure $(\ell+1)$-HIBE scheme. Here, we show it is also possible to obtain CCA security by employing a CCA-secure symmetric key encryption algorithm.

5.1 CCA construction

Our CCA construction resembles the CPA construction in Section 4. As to Setup algorithm, in order to employing symmetric key encryption, we introduce one more collision resistant hash function $H : \mathbb{G}_1 \rightarrow \{0,1\}^n$ to change the message space from $\mathbb{G}_1$ to $\{0,1\}^n$. Here n is the size of message. KeyGen algorithm remains unaltered. To encrypt a message $M \in \{0,1\}^n$ for identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$ with a random number $c \in \mathbb{Z}_p$, we first derive the symmetric key $sk = H(e(P_1, P_2)^c)$ and then use sk to encrypt M. The ciphertext is computed as $C = (A, B, C_1, \ldots, C_j) = (\mathsf{Enc}(sk, M), cP, cF_1(\mathrm{I}_1), \ldots, cF_j(\mathrm{I}_j))$. To decrypt a given ciphertext, we first obtain $e(P_1, P_2)^c$ by computing $e(B, d_0)/\prod_{k=1}^{j} e(C_k, d_k)$, then extract the symmetric key $sk = H(e(P_1, P_2)^c)$ and recover the message $M = \mathsf{Dec}(sk, A)$.

5.2 Security Analysis

We prove the security of our CCA construction under the standard decisional BDH assumption in groups generated by $\mathcal{G}$.

Theorem 5.1 *Our CCA construction is* IND-ID-CCA *assuming the decisional BDH problem is hard in groups generated by $\mathcal{G}_1$. Concretely, suppose there is an* IND-ID-CCA *adversary $\mathcal{A}$ that has advantage ϵ against the scheme. If $\mathcal{A}$ makes at most $q_E > 0$ private key queries and $q_D > 0$ decryption queries, then there is an algorithm $\mathcal{B}$ that solves the decisional BDH generated by $\mathcal{G}$ with advantage at least:*

$$Adv_{\mathcal{B}} \geq \frac{\epsilon}{e(q_E + q_D)^{\ell}}$$

Proof. Suppose $\mathcal{A}$ has advantage ϵ in attacking the CCA construction. We show how to construct an adversary $\mathcal{B}$ that uses adversary $\mathcal{A}$ against decisional BDH problem. Algorithm $\mathcal{B}$ is given as input a random 5-tuple (P, aP, bP, cP, Z) that either sampled from $\mathcal{P}_{BDH}$ (where $Z = e(P,P)^{abc}$) or from $\mathcal{R}_{BDH}$ (where Z is uniform and independent in $\mathbb{G}_1$). Algorithm $\mathcal{B}$'s goal is to output 1 if $Z = e(P,P)^{abc}$ and 0 otherwise. Set $P_1 = aP$, $P_2 = bP$, $P_3 = cP$. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an IND-ID-CPA game as follows.

Setup. The same as CPA construction in Section 4.

Phase 1 - Private key queries. The same as CPA construction in Section 4.

Phase 1 - Decryption queries. Let $C = (A, B, C_1, \ldots, C_u)$ be a decryption query for identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_u)$ issued by algorithm $\mathcal{A}$. $\mathcal{B}$ intends to generate the private key for ID to answer the decryption query. If $l_i = t_i$ holds for any $1 \leq i \leq u$, $\mathcal{B}$ aborts. (In this case, $\mathcal{B}$ cannot generate the private key.) Otherwise, $\mathcal{B}$ generates the private key $d = (d_0, d_1, \ldots, d_u)$ corresponding to ID and obtains $e(P_1, P_2)^c$ by computing $e(B, d_0)/\prod_{i=1}^{u} e(C_i, d_i)$. Then $\mathcal{B}$ returns $\mathsf{Dec}(sk, A)$ to $\mathcal{A}$, where $sk = H(e(P_1, P_2)^c)$.

Challenge. The adversary $\mathcal{A}$ submits two messages $M_0, M_1 \in \{0,1\}^n$ and an target identity $\mathsf{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$, where $1 \leq k \leq \ell$. The only constraint is that ID^* is not a prefix of any identity has been asked for private key in Phase 1. $\mathcal{B}$ first computes $H_1(\mathrm{I}_i^*) = w_i^*$, $H_2(\mathrm{I}_i^*) = l_i^*$. If $l_i^* \neq t_i$ the simulator aborts and submits a random guess for β'. Otherwise, $F_i^*(\mathrm{I}_i^*) = U_i + w_i^* P + k_i^* P_2 = (s_i + w_i^*)P$, $\mathcal{B}$ flips a fair coin β, computes $sk = H(Z)$, and creates the ciphertext as

$$C = (A, B, C_1, \ldots, C_k)$$

where $A = \mathsf{Enc}(sk, M_\beta)$, $B = P_3$, and $C_i = (s_i + w_i^*)P_3$. Note that $C_i = (s_i + w_i^*)P_3 = c(s_i P - t_i P_2 + w_i^* P + l_i^* P_2) = cF_i^*(\mathrm{I}_i^*)$, so we claimed that $(A, B, C_1, \ldots, C_k)$ is a valid ciphertext for ID^* when $Z = e(P_1, P_2)^c = e(P,P)^{abc}$. Otherwise, when Z is a random element of $\mathbb{G}_1$, the ciphertext gives no information about the simulator's choice of β.

Phase 2. $\mathcal{B}$ responds to queries the same way it did in Phase 1.

Guess. Finally, the adversary $\mathcal{A}$ outputs a guess $\beta' \in \{0,1\}$. If $\beta = \beta'$, then $\mathcal{B}$ outputs 1 meaning $Z = e(P,P)^{abc}$. Otherwise, it outputs 0 meaning $Z \neq e(P,P)^{abc}$.

When the input 5-tuple is sampled from $\mathcal{P}_{BDH}$ (where $Z = e(P,P)^{abc}$) then $\mathcal{A}$'s view is identical to its view in a real attack and therefore we have $|\Pr[\beta = \beta'] - 1/2| \geq \epsilon$. On the other hand, when the input 5-tuple is sampled from $\mathcal{R}_{BDH}$ (where Z is uniform in $\mathbb{G}_1$) then $\Pr[\beta = \beta'] = 1/2$. Therefore, with P, U_i uniform in $\mathbb{G}^*$, a, b, c uniform in $\mathbb{Z}_p$, Z uniform in $\mathbb{G}_1$ and assuming SE is $\mathsf{IND}\text{-}\mathsf{CCA}$ secure, we have that

$$\begin{aligned}\big|\Pr[\mathcal{B}(P, P_1, P_2, P_3, e(P,P)^{abc}) = 1] - \Pr[\mathcal{B}(P, P_1, P_2, P_3, Z) = 1]\big| \\ \geq \left|\left(\frac{1}{2} \pm \epsilon\right) - \frac{1}{2}\right| = \epsilon\end{aligned}$$

To complete the proof of Theorem 5.1 it remains to calculate the probability that $\mathcal{B}$ aborts during the simulation. $\mathcal{B}$ may aborts simulation due to the two following events.

1. Event $\mathcal{E}_1$: $\bigvee_{i=1}^{u} l_i = t_i$ for any query in Phase 1 and Phase 2 when answering the private key extraction queries or decryption queries.
2. Event $\mathcal{E}_2$: $\bigvee_{i=1}^{k} l_i^* \neq t_i$ in the challenge phase.

$H_2(\cdot)$ is a collision resistant hash function, its outputs uniformly distribute in $\mathbb{Z}_m$, thus the probability of $\Pr[k_i = t_i]$ is $1/m$. For any private key extraction query for a depth u identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_u)$, the probability that $\mathcal{B}$ can generate the private key is

$$\Pr\left[\bigwedge_{i=1}^{u} l_i \neq t_i\right] = \left(1 - \frac{1}{m}\right)^u \geq \left(1 - \frac{1}{m}\right)^\ell$$

Suppose the maximum number of private key extraction queries is q_E, the maximum number of decryption queires is q_D, then we have

$$\Pr[\neg\mathcal{E}_1] \geq \left(1 - \frac{1}{m}\right)^{\ell(q_E + q_D)}$$

Accordingly to the definition of event $\mathcal{E}_2$, it is easy to see that

$$\Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1] = \left(\frac{1}{m}\right)^k \geq \left(\frac{1}{m}\right)^\ell$$

Therefore

$$\Pr[\mathcal{B} \text{ does not aborts}] = \Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1]\Pr[\neg\mathcal{E}_1] \geq \left(\frac{1}{m}\right)^\ell \left(1 - \frac{1}{m}\right)^{\ell(q_E+q_D)}$$

Using the same technique in Section 4, we can optimize the probability by setting $m_{opt} = 1 + q_E + q_D$. With m_{opt}, we have

$$\Pr[\mathcal{B} \text{ does not aborts}] = \frac{1}{(q_E+q_D)^\ell}\left(1 - \frac{1}{1+q_E+q_D}\right)^{(1+q_E+q_D)\cdot\ell} \approx \frac{1}{e\cdot(q_E+q_D)^\ell}$$

This shows that $\mathcal{B}$'s advantage is at least $\epsilon/(e\cdot(q_E+q_D)^\ell)$ as required. □

6 A Signature scheme

Boneh and Franklin [15] described a generic method for converting any IBE scheme into a signature scheme. The public key of the signature schemes corresponds to the global parameters of the IBE scheme. The signature on a message M is the IBE decryption key for $\mathsf{ID} = M$. To verify a signature, choose a random value M', encrypt M' using the public key $\mathsf{ID} = M$, and then attempt to decrypt using the given signature on M as the decryption key. We note that this is a randomized verification algorithm.

In the generic transformation the security of the resulting signature scheme reduces to the security of the IBE scheme. Thus, we immediately have a signature scheme which is secure assuming the decisional BDH problem. Moreover, we can use the bilinear map in order to deterministically verify a signature and get a signature scheme from our level-1 CPA construction in Section 4 that reduces to the weaker assumption: computational Diffie-Hellman assumption.

6.1 Construction

KeyGen. Run $\mathcal{G}(1^\kappa) \rightarrow (\mathbb{G}, \mathbb{G}_1, p, e)$. Pick a random $a \in \mathbb{Z}_p$ and three random elements $P, P_2, U \in \mathbb{G}$, then compute $P_1 = aP$. Additionally, choose two collision resistant hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_m$. The public key is $PK = (P, P_1, P_2, U, H_1, H_2)$. The private key is $SK = aP_2$.

Signing. Given a private key aP_2, and a message $M \in \{0,1\}^*$, compute $F(M) = U + H_1(M)P + H_2(M)P_2$ and choose a random $r \in \mathbb{Z}_p$. The signature is created as $\sigma_M = (\sigma_1, \sigma_2) = (aP_2 + rF(M), rP)$.

Verification. Given a message $M \in \{0,1\}^*$, and a signature $\sigma = (\sigma_1, \sigma_2) \in \mathbb{G} \times \mathbb{G}$, compute $F(M) = U + H_1(M)P + H_2(M)P_2$ and verify that if $e(\sigma_1, P)/e(\sigma_2, F(M)) = e(P_1, P_2)$ holds. If so, output valid; if not, output invalid.

The verification works because the following equations:

$$\frac{e(\sigma_1, P)}{e(\sigma_2, F(M))} = \frac{e(aP_2 + rF(M), P)}{e(rP, F(M))} = \frac{e(P_1, P_2)e(rF(M), P)}{e(rP, F(M))} = e(P_1, P_2)$$

The verification process can be viewed as a special case of encryption algorithm of the level-1 CPA construction where the random exponentiation $c = 1$.

6.2 Security

We prove the security of our signature scheme against existential forgery under adaptive chosen message attacks in the standard model. Existential unforgeable under a chosen message attack [22] for a signature scheme is defined using the following game between a challenger and an adversary $\mathcal{F}$.

Setup. The challenger runs algorithm KeyGen to obtain a public key PK and private key SK. The adversary $\mathcal{F}$ is given PK.

Signature Queries. Proceeding adaptively, $\mathcal{F}$ requests signatures with PK on at most q_s messages on its choice $M_1, \ldots, M_{q_s} \in \{0,1\}^*$. The challeger responds to each query with a signature $\sigma_{M_i} = \mathsf{Sign}(SK, M_i)$.

Forge. Eventually, $\mathcal{F}$ outputs a pair (M, σ) and wins the game if

1. M is not any of $M_1, \ldots, M_{q_s}$.
2. $\mathsf{Verify}(PK, M, \sigma) =$ valid.

We define $Adv_{\mathcal{F}} = \Pr[\mathsf{Verify}(PK, M, \sigma) =$ valid$]$ to be the probability that $\mathcal{F}$ wins in the above game.

Definition 6.1 *A forger $\mathcal{F}(t, q_s, \epsilon)$ - breaks a signature scheme if $\mathcal{F}$ runs in time at most t, $\mathcal{F}$ makes at most q_s signature queries, and $Adv_{\mathcal{F}}$ is at least ϵ. A signature scheme is (t, q_s, ϵ)-existentially unforgeable under an adaptive chosen message attack if no forger (t, q_s, ϵ) - breaks it.*

Theorem 6.2 *The signature scheme is secure against existential forgery under an adaptive chosen message attack assuming the computational Diffie-Helleman assumption holds. Concretely, if there exists a (t, q_s, ϵ)-forger $\mathcal{F}$ using adaptive chosen message attack for the proposed signature scheme, then there exists an algorithm $\mathcal{B}$ solves computational Diffie-Hellman problem generated by $\mathcal{G}$ with advantage at least:*

$$Adv_{\mathcal{B}} = \frac{\epsilon}{e \cdot q_s}$$

Proof. Suppose $\mathcal{F}$ is a forger algorithm that (t, q_s, ϵ)-breaks the signature scheme. We show how to construct a t'-time algorithm $\mathcal{B}$ that solves computational Diffie-Hellman problem in $\mathbb{G}_1$ with probability at least ϵ'. Algorithm $\mathcal{B}$ is given $P, P_1 = aP, P_2 = bP, e$, note that $a, b \in \mathbb{Z}_p$ are unknown to $\mathcal{B}$. Its goal is to output $abP \in \mathbb{G}_1$. Algorithm $\mathcal{B}$ simulates the challenger and interacts with forger $\mathcal{F}$ as follows.

Setup. The simulator $\mathcal{B}$ randomly picks $s \in \mathbb{Z}_p$, $t \in \mathbb{Z}_m$ and assigns $U = sP - tP_2$. $\mathcal{B}$ starts by giving $\mathcal{F}$ the public key $PK = \langle P, P_1, P_2, U, H_1, H_2 \rangle$. s and t are kept internal to $\mathcal{B}$.

Signature queries. Let M_i be a signature query issued by $\mathcal{F}$. Algorithm $\mathcal{B}$ computes $H_1(M_i) = w_i$, $H_2(M_i) = l_i$. If $l_i = t$, $\mathcal{B}$ reports failure and terminates. (The creation of signature is exactly the generation process of private key. As previous analysis, $\mathcal{B}$ is unable to generate private key when $l_i = t$.) Otherwise, $\mathcal{B}$ computes $Q_i = U + w_iP + l_iP_2$ and choose a random $r \in \mathbb{Z}_p$. The signature is create as

$$\sigma_{M_i} = (\sigma_{1i}, \sigma_{2i}) = \left(r(s + w_i)P + r(l_i - t)P_2 - \frac{s + w_i}{l_i - t}P_1, rP - \frac{1}{l_i - t}P_1 \right)$$

Observe that

$$\begin{aligned}
\sigma_{1i} &= r(s + w_i)P + r(l_i - t)P_2 - \frac{s + w_i}{l_i - t}P_1 \\
&= aP_2 + r(s + w_i)P + r(l_i - t)P_2 - \frac{s + w_i}{l_i - t}aP - aP_2 \\
&= aP_2 + \left(r - \frac{a}{l_i - t} \right) ((s + w_i)P + (l_i - t)P_2) \\
&= aP_2 + \left(r - \frac{a}{l_i - t} \right) (U + w_iP + l_iP_2) \\
&= aP_2 + \tilde{r}Q_i \\
\sigma_{2i} &= rP - \frac{1}{l_i - t}P_1 = \left(r - \frac{a}{l_i - t} \right) P = \tilde{r}P
\end{aligned}$$

We conclude that σ_{M_i} is a valid signature on M_i under the public key (P, P_1, P_2, U) with the underlying random number $\tilde{r} = \left(r - \frac{a}{l_i - t} \right)$.

Forge. At this stage the adversary $\mathcal{F}$ produces a pair (M^*, σ^*) such that no signature query was issued for M^*. Suppose σ^* is a valid signature on M^* under the given public key. $\mathcal{B}$ first computes $H_2(M^*) = l^*$. If $l^* \neq t$, $\mathcal{B}$ reports failure and terminates. Otherwise, $\mathcal{B}$ proceeds on with computing $H_1(M^*) = w^*$, then answers the computational Diffie-Helleman problem as

$$\begin{aligned}
\sigma_1^* - (s + w^*)\sigma_2^* &= aP_2 + r^*(U + w^*P + l^*P_2) - (s + w^*)\sigma_2^* \\
&= aP_2 + r^*((s + w^*)P + (l^* - t)P_2) - (s + w^*)r^*P \\
&= aP_2 = abP
\end{aligned}$$

To complete the proof of Theorem 6.2 it remains to calculate the probability that $\mathcal{B}$ aborts during the simulation. Note that the condition under which $\mathcal{B}$ aborts the game is exactly the same under which the simulator aborts the IND-ID-CPA game in Theorem 4.1. The probability of not aborting is exactly the same as the simulation in Section 4 for $\ell = 1$, thus $\mathcal{B}$'s advantage is at least $\epsilon/(e \cdot q_s)$ as required. □

7 Conclusion

In this paper, we present a variant of Boneh-Boyen BB_1 scheme [6]. By introducing the unbalanced structure, we prove the security of our scheme to be fully secure based on the decisional Bilinear Diffie-Hellman problem without random oracles. Additionally, we showed how to achieve CCA security by employing IND-CCA secure symmetric-key encryption algorithm. Finally, we convert our level-1 CPA construction to an efficient signature scheme that depends only upon the computational Diffie-Hellman assumption in the standard model.

References

1. Shamir, A.: Identity-based cryptosystems and signatures schemes. Advances in Cryptology - Crypto 1984 **196** (1984) 47–53
2. Cocks, C.: An indentity based encryption scheme based on quadratic residues. Institute of Mathematics and Its Applications International Conference on Cryptigraphy and Coding Proceedings of IMA 2001 **2260** (2001) 360–363
3. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. Advances in Cryptology - CRYPTO 2001 **2139** (2001) 213–229
4. Ryuichi Sakai, Kiyoshi Ohgishi, M.K.: Cryptosystems based on pairing. The 2001 Symposium on Cryptography and Information Security, Japan **45** (2001) 26–28
5. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. Advances in Cryptology - Eurocrypt 2003 **2656** (2003) 255–271
6. Boneh, D., Boyen, X.: Efficient selective-id secure identity based encryption without random oracles. Proceedings of Eurocrypt 2004 **3027** (2004) 223–238
7. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. Advances in Cryptology - CRYPTO **3152** (2004) 443–459
8. Waters, B.: Efficient identity-based encryption without random oracles. Advances in Cryptology - EUROCRYPT **3494** (2005) 114–127
9. Naccache, D.: Secure and practical identity-based encryption. IET Inf. Secur. (2005) 59–64
10. Chatterjee, S., Sarkar, P.: Trading time for space: Towards an efficient ibe scheme with short(er) public parameters in the standard model. Information Security and Cryptology (2005) 424–440
11. Gentry, C.: Practical identity-based encryption without random oracles. EUROCRYPT **4004** (2006) 445–464
12. Waters, B.: Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. CRYPTO '09: Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology (2009) 619–636
13. Gentry, C., Silverberg, A.: Hierarchical id-based cryptography. Advances in Cryptology - ASIACRYPT 2002 **2501** (2002) 548–566
14. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. Advances in Cryptology - Eurocrypt 2002 **2322** (2002)
15. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. of Computing **32** (2003) 586–615
16. Halevi, S., Rogaway, P.: A tweakable enciphering mode. **2729** (2003) 482–499
17. Halevi, S., Rogaway, P.: A parallelizable enciphering mode. **2964** (2004) 292–304
18. U.S. Department of Commerce/National Bureau of Standards, National Technical Information Service, Springfield, Virginia. FIPS 197: Advanced encryption standard. Federal Information Processing Standards Publication 197 (2001)

19. Bellare, M., Rogaway, P.: The exact security of digital signatures - how to sign with rsa and rabin. EUROCRYPT (1996) 399–416
20. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identitybased encryption. Advances in Cryptology - Eurocrypt 2004 **3027** (2004) 207–222
21. Boneh, D., Katz, J.: Improved efficiency for cca-secure cryptosystems built using identity-based encryption. (2005) 87–103
22. Goldwasser, S., Micali, S., Rivest, R.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2) (1988) 281–308