Structural Identity-Based Encryption

Man Ho Au[1] and Siu-Ming Yiu[2]

[1] Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia
mhaa456@uow.edu.au

[2] Department of Computer Science
The University of Hong Kong
Pokfulam, Hong Kong
smyiu@cs.hku.hk

Abstract. In this paper, we introduce the concept of structural identity-based encryption (SIBE). Similar to hierarchical identity-based encryption (HIBE), entities in the system are organized into hierarchy. An entity in SIBE can decrypt ciphertext for all its ancestors. It can be seen as an opposite of HIBE, where an entity can decrypt the ciphertext for all its descendants.
We formalize the notion and security requirements, propose an efficient construction and show that our construction is secure under appropriate assumptions in the random oracle model.

Keywords: HIBE, SIBE

1 Introduction

Imagine the following situation. There is an organization O, under which there are several departments and for each department there are several teams. The hierarchical structure of organization O is shown in Fig.1.

We use a vector to represent this hierarchical structure. For example, identity $\texttt{ID}$ of helper H_1 shall be denoted as $(O,\ D_1,\ T_2,\ M_4,\ H_1)$. From time to time messages are encrypted and sent to an entity so that the entity and all its descendants are able to decrypt it. For example, message may be encrypted and sent to team (O, D_1, T_2) and in that case entity with identity (O, D_1, T_2) (think of it as a gatekeeper for team T_2 which inspect all incoming messages), (O, D_1, T_2, M_3), (O, D_1, T_2, M_3, H_1), (O, D_1, T_2, M_3, H_2) shall be able to decrypt the ciphertext.

There is a trusted party, called Key Generation Centre (say, the IT department of organization O), who is responsible for the generation of decryption key $k_{\texttt{ID}}$ for entity with identity $\texttt{ID}$. With $k_{\texttt{ID}}$, entity $\texttt{ID}$ can decrypt all the messages sent to himself and all its ancestors.

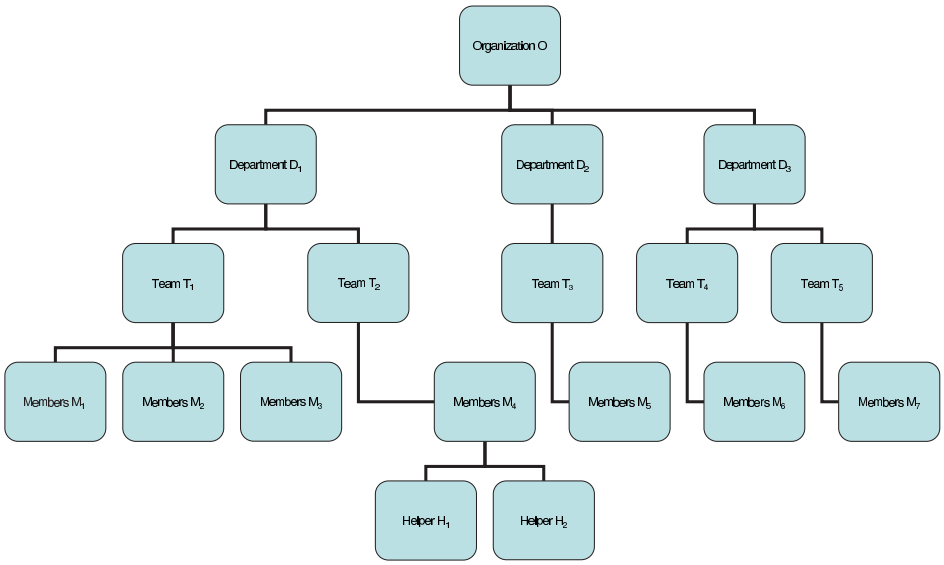

Fig. 1. Structure of Organization O

Naive Approach. A naive solution can be built using identity-based encryption scheme[9]. An entity with identity $\texttt{ID} := (I_1, I_2, \ldots, I_k)$ will be holding the decryption key $d_{\texttt{ID}} := \{d_{I_1}, \ldots, d_{I_k}\}$ such that d_{I_i} is the decryption key for identity $I_1 || \ldots || I_i$ in the identity-based encryption, where $||$ denote string concatenation. To send a message to $\texttt{ID} := (I_1^*, \ldots, I_\ell^*)$, encrypt the message m under the identity $I_1^* || \ldots || I_\ell^*$ of the underlying ID-based encryption.

An obvious problem is that the number of keys required is linear in the depth of the hierarchy. Another problem is that as key (e.g. key for $\texttt{ID} := (O)$) is shared, entity is tempted to leak the key to outsiders without the fear of being identified.

Properties and Requirements. We called a solution to the above problem as structural identity-based encryption system (SIBE). In SIBE, each user is associated with an identity (say $\texttt{ID} := (I_1, \ldots, I_\ell)$). There exists a key generation center (KGC) who is responsible for generating decyrption key for all users. Everyone can encrypt a message m for a target identity $\texttt{ID}^* := (I_1^*, \ldots, I_k^*)$ as ciphertext $ctxt$. All user with identity $\texttt{ID} := (I_1, \ldots, I_\ell)$ can decrypt $ctxt$ with his decryption key if $I_i^* = I_i$ for $i = 1$ to k. We would like to remark that, the encrypter might not know how many entities are there under the target entity. For instance, the encrypter does not need to know the structure of department (O, D_1) to send an encrypted message to (O, D_1).

A secure SIBE should possess *Ciphertext Confidentiality*, *Ciphertext Consistency* and *Key Non-Transferabiilty*. *Ciphertext Confidentiality* refers to the fact

that given a ciphertext, no one can obtain any information about the plaintext. This is similar to the notion of indistinguishability against chosen message and ID attack for IBE. *Ciphertext Consistency* is a new notion regarding SIBE. Following our example of organization O, entity (O, D_1, T_2) is acting as a gatekeeper for team T_2. An adversary may try to send harmful ciphertext to (O, D_1, T_2, M_3) such that it looks harmless if it is decrypted by the key of (O, D_1, T_2). More formally, it is required that no adversary can produce a ciphertext such that the decryption of it using different keys outputs different plaintext. *Key Non-Transferabiilty* prevent a member of the organization from selling his key. More formally, it should be impossible for entity of identity $\texttt{ID} := (I_1, \ldots, I_k)$ to produce a key of identity $\texttt{ID}^* := (I_1, \ldots, I_\ell)$ such that $\ell < k$.[3]

Our Approach. Our construction is a modification of Boneh et al's hierarchical identity-based encryption (HIBE) [1]. We achieve choesen-ciphertext (CCA) security using the ideas in [4] which shows how a CCA secure encryption scheme can be built from weakly-secure (selective-ID, chosen-plain-text-secure) ID-based encryption scheme. The idea of our construction is similar from using a $L + 1$-level CPA-secure HIBE to construct L-level CCA-secure HIBE. In fact, SIBE shares a lot of similarities with HIBE, introduced in [8]. Entities of both SIBE and HIBE are structured within an hierarchy.

In our construction, user key size is constant. However, ciphertext size is linear to the depth of the hierarchy.

Related Results. SIBE is closely related to broadcast encryption (BE), introduced in [7]. The primary difference being the encryptor knows the identity (or public key) of all the recipients in BE while in SIBE, the encryptor might not know the identities within the group of recipients. Nonetheless, the most efficient BE, introduced in [3], achieves $O(1)$ ciphertext size and usr key size. However, public parameter size is of order $O(n)$, where n is the total number of users[4]. A more subtle difference between BE and SIBE is the requirement of *Key Non-Transferabiilty*. Colluding users in BE might be able to produce new decryption key for revoked users. This is demonstrated in an attack on [3] in [12][5].

Our Contributions. We formally introduce the concept of Structural Identity-Based Encryption (SIBE). We define a formal security model to capture the security requirements and provide a concrete construction. We prove that our construction is secure under appropriate assumptions.

Paper Outline. We discuss related works and technical preliminary in the next section. A security model is shown in Section 3. The construction is shown in Section 4, accompanied by security analysis. Finally we conclude in Section 5.

[3] This is in fact a weak requirement. A stronger requirement should be no one is able to produce a decryption blackbox that is not linkable to an identity of the producer. Details of this requirement is discussed in subsequent sections.

[4] A generalized version presented in the same paper achieved a public parameter size of $O(\sqrt{n})$ at a cost of $O(\sqrt{n})$ ciphertext size.

[5] It should be noted that this is not an attack within the security model of BE.

2 Preliminaries

2.1 Notations

We denote $\{0,1\}^*$ the set of bitstrings of length n and $\{0,1\}^*$ the set of bitstrings of arbitrary length. A pattern P is a tuple $(P_1, \ldots, P_\ell) \in (\{0,1\}^*)^\ell$. Let $\mathtt{ID}_1 := (I_1, \ldots, I_{k_1})$, $\mathtt{ID}_2 := (J_1, \ldots, J_{k_2})$ such that $k_1 \geq k_2$ be some patterns. We write $\mathtt{ID}_1 \ni_u \mathtt{ID}_2$ if $I_i = J_i$ for $i = 1$ to k_2. We say $\mathtt{ID}_1$ is under $\mathtt{ID}_2$ if $\mathtt{ID}_1 \ni_u \mathtt{ID}_2$.

2.2 Bilinear Maps

We review the concepts related to bilinear pairings $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

- $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic multiplicative groups of prime order p.
- each element of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ has unique binary representation.
- g_0, h_0 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively.
- ψ is a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(h_0) = g_0$.
- (Bilinear) $\forall x \in \mathbb{G}_1$, $y \in \mathbb{G}_2$ and a, $b \in \mathbb{Z}_p$, $\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}$.
- (Non-degenerate)$\hat{e}(g_0, h_0) \neq 1$.

$\mathbb{G}_1$ and $\mathbb{G}_2$ can be the same or different groups. We say that two groups ($\mathbb{G}_1$, $\mathbb{G}_2$) are a bilinear group pair if the group action in $\mathbb{G}_1$, $\mathbb{G}_2$, the isomorphism ψ and the bilinear mapping $\hat{e}$ are all efficiently computable.

2.3 Mathematical Assumptions

Definition 1 (ℓ-Decisional Bilinear Diffie-Hellman Exponent(Decision ℓ-BDHE) Assumption). *The ℓ-Decisional Bilinear Diffie-Hellman Exponent Assumption (ℓ-BDHE) in bilinear group pair ($\mathbb{G}_1$, $\mathbb{G}_2$) is defined as follows: Given $g \in \mathbb{G}_2$, $h \in \mathbb{G}_2$, g^{α^i} for $i = 1, 2, \cdots, \ell-1, \ell+1, \cdots, 2\ell$, $T \in \mathbb{G}_T$, decide if $T = \hat{e}(\psi(g), h)^{\alpha^\ell}$. We say that the decisional (ℓ, t, ϵ)-BDHE assumption holds in ($\mathbb{G}_1$, $\mathbb{G}_2$) if no t-time algorithm has advantage at least ϵ in solving the decisional ℓ-BDHE problem in ($\mathbb{G}_1$, $\mathbb{G}_2$).*

The decisional ℓ-BDHE assumption was introduced and shown to be hold in in [1] in the generic group model[10].

Definition 2 (ℓ-Computational Bilinear Diffie-Hellman Exponent (Computational ℓ-BDHE) Assumption). *The ℓ-Computational Bilinear Diffie-Hellman Exponent problem is defined in a similar manner as the decisional ℓ-BDHE problem except the solver is now required to output $\hat{e}(\psi(g), h)^{\alpha^\ell}$ (and is not given T as input). We say that the computational (ℓ, t, ϵ)-BDHE assumption holds in ($\mathbb{G}_1$, $\mathbb{G}_2$) if no t-time algorithm has advantage at least ϵ in solving the computational ℓ-BDHE problem in ($\mathbb{G}_1$, $\mathbb{G}_2$).*

2.4 Useful Tools

Message Authentication. One of the building blocks of our system is Message Authentication scheme. Following the notions in [4], a message authentication code is a pair of PPT algorithms (Mac, Vrfy) such that Mac takes as input a key sk and a message m to produce a tag. The algorithm Vrfy takes as input a key sk, a message m and tag and output either `accept` or `reject`. It is required that for all sk and tag, $\mathsf{Vrfy}(sk, m, \mathsf{Mac}(sk, m)) = \texttt{accept}$. Loosely speaking, (Mac , Vrfy) is secure against *one-time chosen-message attack* if no adversary can produce tag', m' such that the following holds:

- The adversary chooses a message m, and is given tag such that $\mathsf{Vrfy}(sk, m, tag) = \texttt{accept}$ for a randomly selected key sk unknown to adversary.
- $\mathsf{Vrfy}(sk, m', tag') = \texttt{accept}$.
- $m \neq m'$ or $tag \neq tag'$.

Encapsulation. Another building block of our system is an Encapsulation scheme, introduced in [4]. Roughly speaking, it is a weak variant of commitment and is defined by a triple of PPT algorithms (Init,S, R) as follow. On input security parameter 1^k, Init output pub. On input a 1^k and pub, S output com, dec and a string $r \in \{0,1\}^k$. On input pub, com, dec, R output r. It is required that for all pub output by Init and for all (r, com, dec) output by $\mathsf{S}(1^k, pub)$, we have $\mathsf{R}(pub, com, dec) = r$. In addition, an encapsulation scheme must satisfy *binding* and *hiding*. Informally speaking, binding means that an honestly generated com can be opened to a single value of r only while hiding means that even given pub and com, the string r should be indistinguishable from random. Very efficient construction (based only on hash function) is given in [4].

3 Model

3.1 Syntax

An SIBE system consists of four algorithms, namely, Setup, KenGen, Encrypt and Decrypt. To represent the hierarchical structure, we follow the notions of hierarchical identity-based encryption system in which each identity `ID` is a vector and a vector of dimension k represents an identity at depth k.

- $\mathsf{Setup}(1^\lambda)$: On input an unary string 1^λ, where λ is a security parameter, the algorithm outputs a master secret key s and public parameter param. In SIBE, the TTP who is responsible for generating keys runs this algorithm, retains s and publishes param.
- Keygen(param, s, `ID`): The algorithm takes as input `ID` and outputs the private key $d_{\texttt{ID}}$ for identity `ID`.
- Encrypt(param, `ID`, m): For a message m together with the target identity `ID`, the algorithm outputs the ciphertext ct which is the encryption of m for identity `ID`.

– $\mathsf{Decrypt}(\mathsf{param}, ct, d_{\mathtt{ID}^*})$: Given the ciphertext ct for identity $\mathtt{ID}$, the private key $d_{\mathtt{ID}^*}$ such that $\mathtt{ID}^*$ is under $\mathtt{ID}$, the algorithm outputs plaintext m.

These algorithms must satisfy the trivial constraint of *correctness*. That is, suppose $(s, \mathsf{param}) \leftarrow \mathsf{Setup}(1^\lambda)$, $ct \leftarrow \mathsf{Encrypt}(\mathsf{param}, \mathtt{ID}, m)$ for any $\mathtt{ID}$ and message m, we require $m \leftarrow \mathsf{Decrypt}(\mathsf{param}, ct, d_{\mathtt{ID}^*})$ for all $\mathtt{ID}^*$ such that $\mathtt{ID}^*$ is under $\mathtt{ID}$ and $d_{\mathtt{ID}^*} \leftarrow \mathsf{KeyGen}(\mathsf{param}, s, \mathtt{ID}^*)$.

Validation of decryption key. Due to a subtlety of defining a security requirement, an SIBE should possess an extra algorithm called $\mathsf{keyValidation}$. $0/1 \leftarrow \mathsf{KeyValidation}(\mathsf{param}, \mathtt{ID}, d_{\mathtt{ID}})$ is run by an entity with identity $\mathtt{ID}$ after he receives decryption key $d_{\mathtt{ID}}$ from KGC. 0 indicates that the key $d_{\mathtt{ID}}$ is valid and 1 indicate that the key is invalid (maybe due to transmission error). It is required that if $0 \leftarrow \mathsf{KeyValidation}(\mathsf{param}, \mathtt{ID}, d_{\mathtt{ID}})$, $m \leftarrow \mathsf{Decrypt}(\mathsf{param}, ct, d_{\mathtt{ID}})$ for all ct and $\mathtt{ID}^*$ such that $\mathtt{ID}$ is under $\mathtt{ID}^*$ and $ct \leftarrow \mathsf{Encrypt}(\mathsf{param}, \mathtt{ID}^*, m)$.

3.2 Security

We use a game-based approach to formally the security requirements of an SIBE. The adversary's capabilities are modeled by arbitrary and adaptive queries to oracles. The oracles are defined as follows.

– Key query ($\mathtt{ID}$): The oracle responds with $d_{\mathtt{ID}}$ such that $d_{\mathtt{ID}}$ is the decryption key of identity $\mathtt{ID}$.
– Decryption query ($\mathtt{ID}, ct$): The oracle responds with plaintext m.
– Hash query $s \in \{0,1\}^*$: The oracle return h which is the hash value of s.

Ciphertext Confidentiality. Ciphertext confidentiality is similar to the standard notion of a secure identity-based encryption system, indistinguishability against adaptive chosen-ciphertext-and-identity attack (IND-ID-CCA)[2]. Precisely, the following game an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ defines *ciphertext confidentiality* of an SIBE system.

Definition 3 (Game IND-ID-CCA).

– (Setup.) *$\mathcal{C}$ chooses a sufficiently large security parameter λ and runs Setup. $\mathcal{C}$ retains s and gives param to $\mathcal{A}$.*
– (Phase 1.) *$\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner.*
– (Challenge.) *$\mathcal{A}$ output an identity $\mathtt{ID}^*$, two messages m_0, m_1 on which it wishes to be challenged. The only restriction is that $\mathcal{A}$ did not previously issue a key query on $\mathtt{ID}^*$ or any key query on $\mathtt{ID}'$ such that $\mathtt{ID}'$ is under $\mathtt{ID}^*$. $\mathcal{C}$ randomly chooses a bit $b \in \{0,1\}$, computes $ct^* = \mathsf{Encrypt}(\mathsf{param}, \mathtt{ID}^*, m_b)$ and sends ct^* to $\mathcal{A}$.*
– (Phase 2.) *$\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner except key query on any $\mathtt{ID}'$ such that $\mathtt{ID}'$ is under or $\mathtt{ID}^*$. $\mathcal{A}$ is also not allowed to issue decryption query on $(ct^*, \mathtt{ID}')$ for any identity $\mathtt{ID}'$ such that $\mathtt{ID}'$ is under $\mathtt{ID}^*$.*

– (Guess.) $\mathcal{A}$ *outputs a guess bit* $b' = \{0, 1\}$. $\mathcal{A}$ *wins the game if* $b = b'$.

The advantage of $\mathcal{A}$ *in Game IND-ID-CCA is defined as the probability that* $\mathcal{A}$ *wins the game minus* $\frac{1}{2}$.

A weaker notion of security regarding identity-based system called selective-ID security, in which the adversary commits ahead of time to the public key it will attack, was introduced in [5, 6]. The game is the same as game IND-ID-CCA except that $\mathcal{A}$ gives $\mathcal{C}$ $\mathtt{ID}^*$ before the *Setup* phase and restrictions on key query in phase 2 applies to phase 1 as well. Similarly, we can define *Selective-ID Ciphertext Confidentiality* with Game IND-sID-CCA such that $\mathcal{A}$ is required to give $\mathcal{C}$ $\mathtt{ID}^*$ before the setup phase.

An SIBE system possesses *Ciphertext Confidentiality* (resp. *Selective-ID Ciphertext Confidentiality*) if no polynomial time adversary has non-negligible advantage in Game IND-ID-CCA (resp. Game IND-sID-CCA).

Ciphertext Consistency. For a given SIBE system, it is required $\mathsf{Decrypt}(\mathsf{param}, ct, d_{\mathtt{ID}_1}) = \mathsf{Decrypt}(\mathsf{param}, ct, d_{\mathtt{ID}_2})$ if the target identity of ct is $\mathtt{ID}$ and both $\mathtt{ID}_1$ and $\mathtt{ID}_2$ are under $\mathtt{ID}$. We require that no adversary, includes adversary who has stolen the master secret from the KGC, can generate a ciphertext that behalf differently under different decryption keys[6]. The following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ formally defines *Ciphertext Consistency.*

Definition 4 (Game Cipher-Consis).

– (Setup.) $\mathcal{C}$ *chooses a sufficiently large security parameter* λ *and runs* Setup. $\mathcal{C}$ *gives* s *and* param *to* $\mathcal{A}$.
– (Phase 1.) $\mathcal{A}$ *can perform a polynomially bounded number of queries to the oracles in an adaptive manner.*
– (End Game.) $\mathcal{A}$ *outputs a ciphertext* $ct^*, \mathtt{ID}^*$, *two identities* $\mathtt{ID}_1$ *and* $\mathtt{ID}_2$ *under* $\mathtt{ID}^*$. *If* $\mathcal{A}$ *never issue key query on* $\mathtt{ID}_1$ *or* $\mathtt{ID}_2$, $\mathcal{C}$ *runs the key query and obtains* $d_{\mathtt{ID}_1}$ *and* $d_{\mathtt{ID}_2}$. $\mathcal{A}$ *wins the game if* $\mathsf{Decrypt}(\mathsf{param}, ct^*, d_{\mathtt{ID}_1}) \neq \mathsf{Decrypt}(\mathsf{param}, ct^*, d_{\mathtt{ID}_2})$.

The advantage of $\mathcal{A}$ *in Game Cipher-Consis is defined as the probability that* $\mathcal{A}$ *wins the game.*

An SIBE system possesses *Ciphertext Consistency* if no polynomial time adversary has non-negligible advantage in Game Cipher-Consis.

Key Non-Transferabiilty. In SIBE, an entity U with identity $\mathtt{ID}$ and key $d_{\mathtt{ID}}$ has the ability to decrypt ciphertext for identity $\mathtt{ID}^*$ if $\mathtt{ID}$ is under $\mathtt{ID}^*$. There is no way to prevent U from lending his key to others. However, what we wish to model is that U cannot generate decryption key $d_{\mathtt{ID}^*}$ of identity $\mathtt{ID}^*$ such that $\mathtt{ID} \ni_u \mathtt{ID}^*$ and $\mathtt{ID} \neq \mathtt{ID}^*$[7]. In that case, U is at a risk of being identified if he is to

[6] Similarly to IBE, KGC is a trusted party in an SIBE system.

[7] If U can generate $d_{\mathtt{ID}^*}$ such that $\mathtt{ID}$ is not under $\mathtt{ID}^*$, he breaks ciphertext confidentiality. If U can generate another $d_{\mathtt{ID}}$, people can identify U as the producing since KGC is trusted and others cannot generate decryption key for $\mathtt{ID}$.

share his decryption power with others. We would like to remark that, however, this requirement is weak as U might be able to construct another decryption algorithm (which might decrypt only some of all possible ciphertext) and output the whole thing as a decryption blackbox. However, it is hard tof formalize the security requirement in that sense and we leave it as an open problem. Nonetheless, the following game defines the notion *Key Non-Transferability* for an SIBE.

Definition 5 (Game Key-No-Trans).

- (Setup.) *$\mathcal{C}$ chooses a sufficiently large security parameter λ and runs* Setup. *$\mathcal{C}$ retains s and gives* param *to $\mathcal{A}$.*
- (Phase 1.) *$\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner.*
- (End Game.) *$\mathcal{A}$ output an identity $\mathtt{ID}^*$ and a decryption key d_{ID^*}. $\mathcal{A}$ wins the game if it never issue key query on $\mathtt{ID}^*$ and* $0 \leftarrow$ *KeyValidation*(param, $\mathtt{ID}^*$, d_{ID^*}).

The advantage of $\mathcal{A}$ in Game Key-No-Trans is defined as the probability that $\mathcal{A}$ wins the game.

An SIBE system possesses *Key Non-Transferability* if no polynomial time adversary has non-negligible advantage in Game Key-No-Trans.

Definition 6. *An SIBE is secure if it possesses* Ciphertext Confidentiality, Ciphertext Consistency *and* Key Non-Transferability.

4 Our Construction

In this section, we describe our cryptographic construction in detail and assess its security.

4.1 System Construction

Let $\mathbb{G}$ be a bilinear group of prime order p and $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map. Let $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$ be a collision resistant hash function. Following the terminology of hierarchical IBE, each entity in the hierarchy is of the form $\mathtt{ID} = (I_1, \cdots, I_\ell)$. The k-th component of the vector $\mathtt{ID}$ corresponds to the identity at level k. Let L be the maximum depth of the hierarchy[8].

Setup. Select $g \in \mathbb{G}$, $\alpha \in \mathbb{Z}_p$ and set $g_1 = g^\alpha$. Pick random elements g_2, g_3, h, h_1, ..., $h_L \in \mathbb{G}$ and set $g_4 = {g_2}^\alpha$.

Pick a secure message authentication code $(\mathsf{Mac}, \mathsf{Vrfy})$. Construction of [11] could be used.

Pick a secure encapsulation scheme $(\mathsf{Init}, \mathsf{S}, \mathsf{R})$. Run $\mathsf{Init}(1^\lambda)$ where λ is the bit length of p and obtain pub. Construction of [4] could be used.

Public parameter is (g_1, g_2, g_3, h, h_1, ..., h_L, pub) and master private key is g_4.

[8] This could be increased dynamically later.

KeyGen. For an entity with identity $\mathtt{ID}_A := (I_1, \ldots, I_k)$, the secret key is $(d_0, d_1, d_2) := (g_4(h_1^{I_1} \cdots h_k^{I_k} g_3)^{r_A}, g^{r_A}, h^{r_A})$ for some random $r_A \in \mathbb{Z}_p$.

KeyValidation. On input $\mathtt{ID} := (I_1, \ldots, I_k)$ and (d_0, d_1, d_2), output 0 if

$$\hat{e}(d_0, g) \stackrel{?}{=} \hat{e}(g_2, g_1)\hat{e}(h_1^{I_1} \cdots h_k^{I_k} g_3, d_1) \text{ and}$$
$$\hat{e}(d_1, h) \stackrel{?}{=} \hat{e}(g, d_2).$$

Output 1 otherwise.

Encrypt. To encrypt a message m for entity with identity $\mathtt{ID} := (I_1, \ldots, I_k)$, run $\mathsf{S}(pub)$ to obtain (r, com, dec) and set $M = m||dec$. Assume there exists a representation of M in $\mathbb{G}_T$[9]. Compute $C_1 = \hat{e}(g_1, g_2)^s M$, $C_2 = g^s$, $C_3 = (h_1^{I_1} \cdots h_k^{I_k} h^{com} g_3)^s, T_{k+1} = h_{k+1}^s, \ldots, T_L = h_L^s$. Let $C = (C_1, C_2, C_3, T_{k+1}, \cdots, T_L)$. Compute $tag = \mathsf{Mac}(r, C)$. The ciphertext $CT = (C, com, tag, \mathtt{ID})$.

Decrypt. Entity with identity $\mathtt{ID}' := (I'_1, \cdots, I'_\ell))$ with decryption key $d_{\mathtt{ID}'} := (d_0, d_1, d_2)$ such that $\mathtt{ID}'$ is under $\mathtt{ID} := (I_1, \ldots, I_k)$ decrypts as follow.

Output invalid ciphertext($\perp$) if any of the following does not hold.

$$\begin{aligned}
\hat{e}(h_1^{I_1} \cdots h_k^{I_k} h^{com} g_3, C_2) &\stackrel{?}{=} \hat{e}(C_3, g), \\
\hat{e}(C_3, h_{k+1}) &\stackrel{?}{=} \hat{e}(h_k^{I_k} h^{com} g_3, T_{k+1}), \\
\hat{e}(T_{k+1}, h_{k+2}) &\stackrel{?}{=} \hat{e}(h_{k+1}, T_{k+2}), \\
&\cdots \\
\hat{e}(T_{L-1}, h_L) &\stackrel{?}{=} \hat{e}(h_{L-1}, T_L).
\end{aligned}$$

Compute $d = d_0 d_2^{com}$. Compute $C' = C_3 T_{k+1}^{I'_{k+1}} \cdots T_\ell^{I'_\ell}$. Compute $M = \frac{C_1 \hat{e}(d_1, C')}{\hat{e}(C_2, d)}$ and obtain m and dec. Compute $r = \mathsf{R}(pub, com, dec)$. If $\mathsf{Vrfy}(r, C, tag) = \mathtt{accept}$, then the plaintext is m, else output invalid ciphertext ($\perp$).

4.2 Security Analysis

Regarding the security of our construction, we have the following theorem.

Theorem 1. *Our proposed construction is secure under the Decisional BDHE assumption in the random oracle model.*

[9] In practical scenario, a random element Q in $\mathbb{G}_T$ could be chosen and $H(Q)$ is used as a session key which is used to encrypt $m||dec$.

Ciphertext Confidentiality. We first proof the following lemma which states that our scheme possesses Selective-ID Ciphertext Confidentiality. Next we outline how our scheme possesses Ciphertext Confidentiality in the random oracle model.

Lemma 1. *Our construction possesses Selective-ID Ciphertext Confidentiality under the Decisional BDHE assumption in the standard model.*

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ with non-negligible advantage in Game IND-sID-CCA. Suppose the adversary chooses target identity $\texttt{ID}^* = (I_1^*, \ldots, I_k^*)$, we construct a simulator $\mathcal{S}$ that solves the decisional $(k+2)$-BDHE problem.

Setup. $\mathcal{S}$ receives a problem instance $(g, g_0, g^{\alpha^1}, \ldots, g^{\alpha^{k+1}}, g^{\alpha^{k+3}}, \ldots, g^{\alpha^{2k+4}}, T)$. $\mathcal{S}$'s goal is to decide if $T = \hat{e}(g, g_0)^{\alpha^{k+2}}$. $\mathcal{S}$ generates the system parameter of the SIBE scheme by selecting a random $\gamma \in \mathbb{Z}_p$ and computes $g_1 = g^{\alpha}$ and $g_2 = g^{\gamma + (\alpha^{k+1})}$. $\mathcal{S}$ then randomly picks $\gamma_1, \ldots, \gamma_L, \gamma_h \in_R \mathbb{Z}_p$ and sets $h_i = g^{\gamma_i - \alpha^{(k+1-i)}}$ for $i = 1$ to k and $h_i = g^{\gamma_i}$ when $i = k+1, \ldots, L$ and $h = g^{\gamma_h - \alpha^{k+1}}$.

$\mathcal{S}$ chooses a secure encapsulation scheme $(\mathsf{Init}, \mathsf{S}, \mathsf{R})$, runs Init to obtain pub and then runs $\mathsf{S}(pub)$ to obtain (r^*, com^*, dec^*). $\mathcal{S}$ chooses a secure message authentication scheme $(\mathsf{Mac}, \mathsf{Vrfy})$.

$\mathcal{S}$ randomly selects $\delta \in_R \mathbb{Z}_p$ and sets $g_3 = g^{\delta + \sum_{i=1}^{k} (I_i^* \alpha^{k+1-i}) + com^* \alpha^{k+1}}$. Note that $g_4 = g_2^{\alpha} = g^{\gamma\alpha} g^{\alpha^{k+2}}$ is unknown to $\mathcal{S}$.

$\mathcal{S}$ gives $\mathcal{A}$ the system parameters $(g, g_1, g_2, g_3, h, h_1, \ldots, h_L)$, the encapsulation scheme and message authentication scheme.

Phase 1 & 2. During phase 1 and 2, $\mathcal{A}$ may consults $\mathcal{S}$ for oracle queries in an adaptive manner. Simulation is shown below and we first outline how to construct the challenge ciphertext in the challenge phase.

Challenge Phase. When $\mathcal{S}$ receives m_0, m_1 from $\mathcal{A}$, it randomly selects a bit b and set $M = m_b || dec^*$. It sets $C_1 = M T \hat{e}(g^{\alpha}, h^{\gamma})$, $C_2 = g_0$, $C_3 = g_0^{\delta + \sum_{i=1}^{k} (I_i^* \gamma_i) + com^* \gamma_h}$, $T_{k+i} = g_0^{\gamma_{k+i}}$ (for $i = 1$ to $L-k$), $C = (C_1, C_2, C_3, T_{k+1}, \ldots, T_L)$, $tag = \mathsf{Mac}(r, C)$. The ciphertext is $CT^* = (C, com^*, tag, \texttt{ID}^*)$. It is straight forward to show that CT^* is a valid ciphertext if $T = \hat{e}(g, g_0)^{\alpha^{k+2}}$ and CT^* is independent to b otherwise.

Key Queries. Key query on $(I_1, \ldots, I_n)$ can be divided into 3 types:

1. $(n < k.)$ Let $r = \tilde{r} - \frac{\alpha^{k+1}}{I_k^*}$ for a randomly selected $\tilde{r} \in_R \mathbb{Z}_p$.

$$
\begin{aligned}
d_0 &= g_4\big(g_3 {h_1}^{I_1} \cdots {h_n}^{I_n}\big)^r \\
&= g^{\left(\gamma\alpha+\alpha^{k+2}\right)} g^{r\left[\delta+\sum_{i=1}^{k}(I_i^*\alpha^{k+1-i})+com^*(\alpha^{k+1})+\sum_{i=1}^{n}(I_i(\gamma_i-\alpha^{(k+1-i)}))\right]} \\
&= g^{\left(\gamma\alpha+\alpha^{k+2}\right)} g^{r\left[\delta+\alpha^{k+1}com^*+\sum_{i=1}^{n}(I_i\gamma_i)+\sum_{i=1}^{n}((I_i^*-I_i)\alpha^{k+1-i})\right]} \\
&\quad g^{r\left[\sum_{i=n+1}^{k-1}(I_i^*\alpha^{k+1-i})+I_k^*\alpha\right]} \\
&= \left(g^{\alpha}\right)^{\gamma}\left[\left(g^{\delta}\right)\left(g^{\alpha^{k+1}}\right)^{com^*}\left(\prod_{i=1}^{n} g^{I_i\gamma_i}\right)\left(\prod_{i=1}^{n}(g^{\alpha^{k+1-i}})^{I_i^*-I_i}\right)\right]^{\tilde{r}} \\
&\quad \left[\left(\prod_{i=n+1}^{k-1}(g^{\alpha^{k+1-i}})^{I_i^*}\right)\left(g^{\alpha}\right)^{I_k^*}\right]^{\tilde{r}}\left[\left(g^{\alpha^{k+1}}\right)^{\delta}\left(g^{\alpha^{2k+2}}\right)^{com^*}\right]^{\frac{-1}{I_k^*}} \\
&\quad \left[\left(\prod_{i=1}^{n}(g^{\alpha^{k+1}})^{I_i\gamma_i}\right)\left(\prod_{i=1}^{n}(g^{\alpha^{2k+2-i}})^{I_i^*-I_i}\right)\left(\prod_{i=n+1}^{k-1}(g^{\alpha^{2k+2-i}})^{I_i^*}\right)\right]^{\frac{-1}{I_k^*}}
\end{aligned}
$$

Note that d_0 does not involve the term $g^{\alpha^{k+2}}$ and is thus computable by $\mathcal{S}$. Compute $d_1 = g^{\tilde{r}}(g^{\alpha^{k+1}})^{\frac{-1}{I_k^*}}$ and $d_2 = h^{\tilde{r}}\ [(g^{\alpha^{k+1}})^{\gamma_h}(g^{-\alpha^{2k+2}})]^{\frac{1}{I_k^*}}$. $\mathcal{S}$ returns (d_0, d_1, d_2) to $\mathcal{A}$.

2. $(n = k.)$ Let m be biggest index such that $I_m \neq I_m^*$. Such m always exists because $\mathcal{A}$ is not allowed to query the target identity. Let $r = \tilde{r} + \frac{\alpha^{m+1}}{I_m - I_m^*}$ for a randomly selected $\tilde{r} \in_R \mathbb{Z}_p$.

$$
\begin{aligned}
d_0 &= g_4\big(g_3 {h_1}^{I_1} \cdots {h_k}^{I_k}\big)^r \\
&= g^{\left(\gamma\alpha+\alpha^{k+2}\right)} g^{r\left[\delta+\sum_{i=1}^{k}(I_i^*\alpha^{k+1-i})+com^*(\alpha^{k+1})+\sum_{i=1}^{k}(I_i(\gamma_i-\alpha^{(k+1-i)}))\right]} \\
&= g^{\left(\gamma\alpha+\alpha^{k+2}\right)} g^{r\left[\delta+\alpha^{k+1}com^*+\sum_{i=1}^{k}(I_i\gamma_i)\right]} \\
&\quad g^{r\left[\sum_{i=1,i\neq m}^{k}((I_i^*-I_i)\alpha^{k+1-i})+(I_m^*-I_m)\alpha^{k+1-m}\right]} \\
&= \left(g^{\alpha}\right)^{\gamma}\left[\left(g^{\delta}\right)\left(g^{\alpha^{k+1}}\right)^{com^*}\left(\prod_{i=1}^{n} g^{I_i\gamma_i}\right)\right]^{\tilde{r}} \\
&\quad \left[\left(\prod_{i=1,i\neq m}^{n}(g^{\alpha^{k+1-i}})^{I_i^*-I_i}\right)\left(g^{\alpha^{k+1-m}}\right)^{I_m^*-I_m}\right]^{\tilde{r}} \\
&\quad \left[\left(g^{\alpha^{m+1}}\right)^{\delta}\left(g^{\alpha^{k+m+2}}\right)^{com^*}\left(\prod_{i=1}^{n}(g^{\alpha^{m+1}})^{I_i\gamma_i}\right)\right]^{\frac{1}{I_m-I_m^*}} \\
&\quad \left[\left(\prod_{i=1,i\neq m}^{n}(g^{\alpha^{k+m+2-i}})^{I_i^*-I_i}\right)\right]^{\frac{1}{I_m-I_m^*}}
\end{aligned}
$$

Again, d_0 does not involve the term $g^{\alpha^{k+2}}$. Compute $d_1 = g^{\tilde{r}}(g^{\alpha^{m+1}})^{\frac{1}{I_m-I_m^*}}$ and $d_2 = h^{\tilde{r}}[(g^{\alpha^{m+1}})^{\gamma_h}(g^{-\alpha^{k+m+2}})]^{\frac{1}{I_m-I_m^*}}$. $\mathcal{S}$ returns (d_0, d_1, d_2) to $\mathcal{A}$.

3. ($n > k$.) Since $\mathcal{A}$ is not allowed to query $(I_1^*, \ldots, I_k^*)$ or any identity under $\mathtt{ID}^*$, $\mathcal{S}$ can obtain the secret key of $(I_1, \cdots, I_k)$ using the methods above and let the resulting secret key be (d_0, d_1, d_2). Then $\mathcal{S}$ returns $(d_0 {d_1}^{\gamma_{k+1} I_{k+1}} \cdots {d_1}^{\gamma_n I_n}, d_1, d_2)$ to $\mathcal{A}$.

Decryption Queries. $\mathcal{S}$ can simulate decryption queries not related to the target identity perfectly since it possesses decryption keys of those identities. We show how $\mathcal{S}$ simulate decryption query of ciphertext $CT = (C, com, tag, \mathtt{ID})$ for the identity $\mathtt{ID} = \mathtt{ID}^*$ or any identities under $\mathtt{ID}^*$ ($\mathtt{ID} := (I_1^*, \ldots, I_k^*, I_{k+1}, \ldots, I_n)$).

If $com = com^*$, return invalid ciphertext. Probability of rejecting valid ciphertext is negligible by the following arguments. Before the challenge phase, since com^* is unknown to $\mathcal{A}$, probability that $\mathcal{A}$ submit $com = com^*$ is negligible. After $\mathcal{A}$ receives the challenge ciphertext, probability that $\mathcal{A}$ could generate a valid ciphertext not equal to the challenge ciphertext with $com = com^*$ is negligible if the encapsulation scheme and the message authentication scheme are secure.

Compute $d = g_4\big(g_3 {h_1}^{I_1^*} \cdots {h_k}^{I_k^*} h^{com} h_{k+1}^{I_{k+1}} \cdots h_n^{I_n}\big)^r$, where $r = \tilde{r} + \frac{\alpha}{com - com^*}$ for a randomly selected $\tilde{r} \in_R \mathbb{Z}_p$, as follow.

$$
\begin{aligned}
d &= g_4\big(g_3 {h_1}^{I_1^*} \cdots {h_k}^{I_k^*} h^{com} h_{k+1}^{I_{k+1}} \cdots h_n^{I_n}\big)^r \\
&= g^{\left(\gamma\alpha + \alpha^{k+2}\right)} g^{r\left[\delta + \sum_{i=1}^{k}(I_i^* \alpha^{k+1-i}) + com^*(\alpha^{k+1})\right]} \\
&\quad g^{r\left[+\sum_{i=1}^{k}(I_i^*(\gamma_i - \alpha^{(k+1-i)})) + com(\gamma_h - \alpha^{k+1}) + \sum_{i=k+1}^{n}(I_i\gamma_i)\right]} \\
&= g^{\left(\gamma\alpha + \alpha^{k+2}\right)} g^{r\left[\delta + \alpha^{k+1}(com^* - com) + \sum_{i=1}^{k}(I_i^*\gamma_i) + com\gamma_h + \sum_{i=k+1}^{n}(I_i\gamma_i)\right]} \\
&= \left(g^{\alpha}\right)^{\gamma} \left[\big(g^{\delta}\big)\big(g^{\alpha^{k+1}}\big)^{(com^* - com)}\Big(\prod_{i=1}^{k} g^{I_i^*\gamma_i}\Big)\Big(\prod_{i=k+1}^{n} g^{I_i\gamma_i}\Big)\right]^{\tilde{r}} \\
&\quad \left[\big(g^{\alpha}\big)^{\delta}\Big(\prod_{i=1}^{k}(g^{\alpha})^{I_i * \gamma_i}\Big)\Big(\prod_{i=k+1}^{n}(g^{\alpha})^{I_i\gamma_i}\Big)\right]^{\frac{1}{com - com^*}}
\end{aligned}
$$

Compute $d_1 = g^{\tilde{r}} (g^{\alpha})^{\frac{1}{com - com^*}}$. $\mathcal{S}$ then computes M by $\frac{C_1 \hat{e}(d_1, C_3)}{\hat{e}(C_2, d)}$. Parse M and obtain m and dec. Then compute $r = \mathsf{R}(pub, com, dec)$ and if $\mathsf{Vrfy}(r, CT, tag) = \mathtt{accept}$, return m, else return invalid ciphertext.

Answer If $\mathcal{A}$ can guess correctly the bit b, the $\mathcal{S}$ answer that $T = \hat{e}(g, h)^{\alpha^{k+2}}$ and no otherwise. It is easy to verify that advantage of $\mathcal{A}$ in wining is the advantage of $\mathcal{S}$ in solving the hard problem.

Ciphertext Confidentiality. Ciphertext Confidentiality can be achieved by standard argument in the random oracle model when each I_i is hashed first before use. It introduces a $\frac{1}{q_H^L}$ term in the reduction, where q_H is the total number of hash queries.

Ciphertext Consistency. It is straight-forward to show that our scheme possesses *Ciphertext Consistency*. Let $(C_1, C_2, C_3, T_{k+1}, \ldots, T_L)$ be a ciphertext for identity $\mathtt{ID} := (I_1, \ldots, I_k)$. The checking during the decryption ensure the DL of $C_2, C_3, T_{k+1}, \ldots, T_L$ to their corresponding bases are the same. Likewise, let $\mathtt{ID}' := (I_1, \ldots, I_k, I'_{k+1}, \ldots, I'_n)$, $\mathtt{ID}^* := (I_1, \ldots, I_k, I^*_{k+1}, \ldots, I^*_m)$ be identities under $\mathtt{ID}$ such that (d'_0, d'_1, d'_2) and (d^*_0, d^*_1, d^*_2) are the corresponding decryption key. Due to the setting of the game, these two decryption key must be correctly formed. Let $d' = d'_0 {d'_2}^{com}$ and $d^* = d^*_0 {d^*_2}^{com}$ and $C' = C_3 T_{k+1}^{I'_{k+1}} \cdots T_n^{I'_n}$ and $C'^* := C_3 T_{k+1}^{I^*_{k+1}} \cdots T_m^{I^*_m}$.

Values of $\frac{\hat{e}(d'_1, C')}{\hat{e}(C_2, d')}$ and $\frac{\hat{e}(d^*_1, C'^*)}{\hat{e}(C_2, d^*)}$ are the same. Thus, it is impossible to compute a ciphertext which breaks the ciphertext consistency.

Key Non-Transferability. Using similar argument of transforming SIBE with Selective-ID Ciphertext Confidentiality into SIBE with Ciphertext Confidentiality, we assume $\mathcal{S}$ correctly guess the challenge identity in Game Key-No-Trans in the random oracle model. Let L be the maximum depth of the hierarchy. Assume $\mathtt{ID}^* := (I^*_1, \ldots, I^*_k)$ be the identity output by the adversary. If an adversary could win in Game Key-No-Trans with identity $\mathtt{ID}^*$ without key queries to any identity under it, it could win in Game IND-ID-CCA. Thus, we assume $1 \leq k < L$. Consequently, probability that $\mathcal{S}$ guess correctly guess $\mathtt{ID}$, in the random oracle model, is $\left(\frac{1}{q_H}\right)^{L-1}$, where q_H is the number of hash queries.

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ with non-negligible advantage in Game Key-No-Trans. Suppose $\mathcal{A}$'s final output identity is $\mathtt{ID}^* := (I^*_1, \ldots, I^*_k)$, we construct a simulator $\mathcal{S}$ that solves the computational $(L+2)$-BDHE problem.

Setup. $\mathcal{S}$ receives a problem instance $(g, g_0, g^{\alpha^1}, \ldots, g^{\alpha^{L+1}}, g^{\alpha^{L+3}}, \ldots, g^{\alpha^{2L+4}})$ and is required to output $T = \hat{e}(g, g_0)^{\alpha^{L+2}}$. $\mathcal{S}$ generates the system parameter of the SIBE scheme by selecting a random $\gamma \in \mathbb{Z}_p$ and sets $g_1 = g^\alpha$ and $g_2 = g^{\gamma + (\alpha^{L+1})}$. $\mathcal{S}$ then randomly picks $\gamma_1, \ldots, \gamma_L, \gamma_h \in_R \mathbb{Z}_p$ and sets $h_i = g^{\gamma_i - \alpha^{(L+1-i)}}$ for $i = 1$ to L and $h = g^{\gamma_h - \alpha^{L+1}}$.

$\mathcal{S}$ chooses a secure encapsulation scheme $(\mathsf{Init}, \mathsf{S}, \mathsf{R})$ and a secure message authentication scheme $(\mathsf{Mac}, \mathsf{Vrfy})$.

$\mathcal{S}$ randomly selects $\delta \in_R \mathbb{Z}_p$ and sets $g_3 = \left(g^\delta\right)\left(h_1^{I^*_1} \cdots h_k^{I^*_k}\right)^{-1}$. Note that $g_3 = g^{\delta + \sum_{i=1}^{k} (I^*_i \alpha^{L+1-i} - I^*_i \gamma_i)}$. Also, the master secret key $g_4 = g_2^\alpha = g^{\gamma\alpha} g^{\alpha^{L+2}}$ is unknown to $\mathcal{S}$.

$\mathcal{S}$ gives $\mathcal{A}$ the system parameters (g, g_1, g_2,g_3, h, h_1, …, h_L), the encapsulation scheme and message authentication scheme.

Phase 1. During phase 1, $\mathcal{A}$ may consults $\mathcal{S}$ for oracle queries in an adaptive manner. In particular, $\mathcal{A}$ can ask for all keys in the hierarchy except $\mathtt{ID}^*$. We show how $\mathcal{S}$ responds to key query first.

Key Queries. Similar to the case in Game IND-sID-CCA, key query on $\mathtt{ID} := (I_1, \ldots, I_n)$ can be divided into 3 types:

1. ($n < k$.) Let $r = \tilde{r} - \frac{\alpha^{k+1}}{I_k^*}$ for a randomly selected $\tilde{r} \in_R \mathbb{Z}_p$.

$$
\begin{aligned}
d_0 &= g_4\big(g_3 {h_1}^{I_1} \cdots {h_n}^{I_n}\big)^r \\
&= g^{\left(\gamma\alpha+\alpha^{L+2}\right)} g^{r\left[\delta+\sum_{i=1}^{k}(I_i^*\alpha^{L+1-i}-I_i^*\gamma_i)+\sum_{i=1}^{n}(I_i(\gamma_i-\alpha^{(L+1-i)}))\right]} \\
&= g^{\left(\gamma\alpha+\alpha^{L+2}\right)} g^{r\left[\delta+\sum_{i=1}^{n}((I_i-I_i^*)\gamma_i)+\sum_{i=1}^{n}((I_i^*-I_i)\alpha^{L+1-i})\right]} \\
&\quad g^{r\left[\sum_{i=n+1}^{k-1}(I_i^*\alpha^{L+1-i}-I_i^*\gamma_i)-I_k^*\gamma_k+I_k^*\alpha^{L+1-k}\right]} \\
&= \left(g^{\alpha}\right)^{\gamma}\left[\big(g^{\delta}\big)\Big(\prod_{i=1}^{n} g^{(I_i-I_i^*)\gamma_i}\Big)\Big(\prod_{i=1}^{n}(g^{\alpha^{L+1-i}})^{I_i^*-I_i}\Big)\right]^{\tilde{r}} \\
&\quad \left[\Big(\prod_{i=n+1}^{k-1}(g^{\alpha^{L+1-i}-\gamma_i})^{I_i^*}\Big)\big(g^{\alpha^{L+1-k}-\gamma_k}\big)^{I_k^*}\right]^{\tilde{r}}\left[\big(g^{\alpha^{k+1}}\big)^{-\delta}\big(g^{\alpha^{k+1}\gamma_k}\big)\right]^{\frac{1}{I_k^*}} \\
&\quad \left[\Big(\prod_{i=1}^{n}(g^{\alpha^{k+1}})^{(I_i^*-I_i)\gamma_i}\Big)\Big(\prod_{i=1}^{n}(g^{\alpha^{L+k+2-i}})^{I_i-I_i^*}\Big)\Big(\prod_{i=n+1}^{k-1}(g^{\alpha^{L+k+2-i}})^{-I_i^*}\Big)\right]^{\frac{1}{I_k^*}} \\
&\quad \left[\Big(\prod_{i=n+1}^{k-1}(g^{\alpha^{k+1}})^{I_i^*\gamma_i}\Big)\right]^{\frac{1}{I_k^*}}
\end{aligned}
$$

Note that d_0 does not involve the term $g^{\alpha^{L+2}}$ and is thus computable by $\mathcal{S}$. Compute $d_1 = g^{\tilde{r}}(g^{\alpha^{L+1}})^{\frac{-1}{I_k^*}}$ and $d_2 = h^{\tilde{r}}\ [(g^{\alpha^{L+1}})^{\gamma_h}(g^{-\alpha^{2L+2}})]^{\frac{1}{I_k^*}}$. $\mathcal{S}$ returns (d_0, d_1, d_2) to $\mathcal{A}$.

2. ($n = k$.) Let m be biggest index such that $I_m \neq I_m^*$. Such m always exists because $\mathcal{A}$ is not allowed to query $\mathtt{ID}^*$. Let $r = \tilde{r} + \frac{\alpha^{m+1}}{I_m - I_m^*}$ for a randomly selected $\tilde{r} \in_R \mathbb{Z}_p$.

$$
\begin{aligned}
d_0 &= g_4\big(g_3 {h_1}^{I_1} \cdots {h_k}^{I_k}\big)^r \\
&= g^{\left(\gamma\alpha+\alpha^{L+2}\right)} g^{r\left[\delta+\sum_{i=1}^{k}(I_i^*\alpha^{L+1-i}-I_i^*\gamma_i)+\sum_{i=1}^{k}(I_i(\gamma_i-\alpha^{(L+1-i)}))\right]} \\
&= g^{\left(\gamma\alpha+\alpha^{L+2}\right)} g^{r\left[\delta+\sum_{i=1}^{k}((I_i-I_i^*)\gamma_i)\right]} \\
&\quad g^{r\left[\sum_{i=1,i\neq m}^{k}((I_i^*-I_i)\alpha^{L+1-i})+(I_m^*-I_m)\alpha^{L+1-m}\right]} \\
&= \left(g^{\alpha}\right)^{\gamma}\left[\big(g^{\delta}\big)\Big(\prod_{i=1}^{k} g^{(I_i-I_i^*)\gamma_i}\Big)\right]^{\tilde{r}} \\
&\quad \left[\Big(\prod_{i=1,i\neq m}^{k}(g^{\alpha^{L+1-i}})^{I_i^*-I_i}\Big)\big(g^{\alpha^{L+1-m}}\big)^{I_m^*-I_m}\right]^{\tilde{r}} \\
&\quad \left[\big(g^{\alpha^{m+1}}\big)^{\delta}\Big(\prod_{i=1}^{k}(g^{\alpha^{m+1}})^{(I_i-I_i^*)\gamma_i}\Big)\Big(\prod_{i=1,i\neq m}^{k}(g^{\alpha^{L+m+2-i}})^{I_i^*-I_i}\Big)\right]^{\frac{1}{I_m-I_m^*}}
\end{aligned}
$$

Again, d_0 does not involve the term $g^{\alpha^{L+2}}$. Compute $d_1 = g^{\tilde{r}}(g^{\alpha^{m+1}})^{\frac{1}{I_m-I_m^*}}$ and $d_2 = h^{\tilde{r}}[(g^{\alpha^{m+1}})^{\gamma_h}(g^{-\alpha^{L+m+2}})]^{\frac{1}{I_m-I_m^*}}$. $\mathcal{S}$ returns (d_0, d_1, d_2) to $\mathcal{A}$.

3. ($n > k$.) Let $r = \tilde{r} + \frac{\alpha^{n+1}}{I_n}$ for a randomly selected $\tilde{r} \in_R \mathbb{Z}_p$.

$$
\begin{aligned}
d_0 &= g_4\big(g_3 {h_1}^{I_1} \cdots {h_n}^{I_n}\big)^r \\
&= g^{\left(\gamma\alpha+\alpha^{L+2}\right)} g^{r\left[\delta+\sum_{i=1}^{k}(I_i^*\alpha^{L+1-i}-I_i^*\gamma_i)+\sum_{i=1}^{n}(I_i(\gamma_i-\alpha^{(L+1-i)}))\right]} \\
&= g^{\left(\gamma\alpha+\alpha^{L+2}\right)} g^{r\left[\delta+\sum_{i=1}^{k}((I_i^*-I_i)\alpha^{L+1-i})\right]} \\
&\quad g^{r\left[\sum_{i=1}^{k}((I_i-I_i^*)\gamma_i)+\sum_{i=k+1}^{n}(I_i\gamma_i)-\sum_{i=k+1}^{n}(I_i\alpha^{L+1-i})\right]} \\
&= \left(g^{\alpha}\right)^{\gamma}\left[\big(g^{\delta}\big)\Big(\prod_{i=1}^{k} g^{(I_i-I_i^*)\gamma_i}\Big)\Big(\prod_{i=1}^{k}(g^{\alpha^{L+1-i}})^{I_i^*-I_i}\Big)\right]^{\tilde{r}} \\
&\quad \left[\Big(\prod_{i=k+1}^{n}(g^{I_i\gamma_i})\Big)\Big(\prod_{i=k+1}^{n}(g^{\alpha^{L+1-i}})^{-I_i}\Big)\right]^{\tilde{r}} \\
&\quad \left[\big(g^{\alpha^{n+1}}\big)^{\delta}\Big(\prod_{i=1}^{k}(g^{\alpha^{L+n+2-i}})^{(I_i^*-I_i)}\Big)\Big(\prod_{i=1}^{k}(g^{\alpha^{n+1}})^{(I_i-I_i^*)\gamma_i}\Big)\right]^{\frac{1}{I_n}} \\
&\quad \left[\Big(\prod_{i=k+1}^{n}(g^{\alpha^{n+1}})^{I_i\gamma_i}\Big)\Big(\prod_{i=k+1}^{n-1}(g^{\alpha^{L+n+2-i}})^{I_i}\Big)\right]^{\frac{1}{I_n}}
\end{aligned}
$$

Decryption Queries. $\mathcal{S}$ can simulate decryption queries for all identities perfectly except $\mathtt{ID}^*$ since it possesses decryption keys of those identities. For decryption queries related to $\mathtt{ID}^*$, $\mathcal{S}$ make use of the decryption key of any identities under $\mathtt{ID}^*$. Due to *Ciphertext Consistency* of our scheme, the simulation is perfect.

Answer Finally, $\mathcal{A}$ output a decryption key (d_0^*, d_1^*, d_2^*) of $\mathtt{ID}^*$ such that

$$
\begin{aligned}
\hat{e}(d_0^*, g) =& \hat{e}(g_2, g_1)\hat{e}(h_1^{I_1^*} \cdots h_k^{I_k^*} g_3, d_1^*) \text{ and} \\
\hat{e}(d_1^*, h) =& \hat{e}(g, d_2^*).
\end{aligned}
$$

From the first equation, $\hat{e}(d_0^*, g) = \hat{e}(g_2, g_1)\hat{e}(g^{\delta}, d_1^*)$. Thus, $\hat{e}(d_0^*(d_1^*)^{-\delta}, g) = \hat{e}(g_2, g_1)$. $\mathcal{S}$ returns $\hat{e}(d_0^*(d_1^*)^{-\delta} g^{-\alpha\gamma}, g_0)$ as the answer of the computational $(L+2)$-BDHE problem.

This completes the proof of Theorem 1. □

5 Conclusion

We introduce a new notion called SIBE. We define security model and propose an efficient construction. Our construction is secure under the BDHE assumption in the random oracle model.

References

1. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

2. Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, 2001.
3. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pages 258–275, 2005.
4. Dan Boneh and Jonathan Katz. Improved efficiency for cca-secure cryptosystems built using identity-based encryption. In *CT-RSA*, pages 87–103, 2005.
5. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.
6. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.
7. Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO*, pages 480–491, 1993.
8. Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT*, pages 466–481, 2002.
9. Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53, 1984.
10. Victor Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *EUROCRYPT*, pages 256–266, 1997.
11. Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.
12. Qianhong Wu, Willy Susilo, Yi Mu, and Bo Qin. Cryptanalysis of bgw broadcast encryption schemes for dvd content protection. In *ATC*, pages 32–41, 2007.