Identity-based Hierarchical Designated Decryption*

SHU-HUI CHANG[1], CHUAN-MING LI[2] AND TZONELIH HWANG[3]

[1]*Center of General Education*
Southern Taiwan University of Technology
Tainan, 710 Taiwan
[2]*Department of Information Management*
Shu-Zen College of Medicine and Management
Kaohsiung, 821 Taiwan
[3]*Department of Computer Science and Information Engineering*
National Cheng Kung University
Tainan, 701 Taiwan

This paper presents an identity-based hierarchical designated decryption (IHDD) scheme which allows a message sender to generate ciphertexts that can be decrypted by (1) only a specified recipient or (2) a specified recipient and all or some of its ancestor users in the hierarchy tree. The newly proposed scheme can be considered as a combination of the hierarchical identity-based encryption (HIBE) and the identity-based multi-recipient encryption scheme (ID-based MRES). However, the purpose and structure of the proposed IHDD scheme are different from those of the HIBE and the ID-based MRES. The proposed IHDD scheme has low computation complexity, in which the decryption operation needs only one bilinear pairing computation, and constant length private keys wherein the length of users' private keys is independent of the hierarchy depth.

The security of the proposed scheme is based on the decision bilinear Diffie-Hellman inversion assumption without using random oracles.

Keywords: data security, hierarchical, identity-based encryption, key escrow, multi-recipient encryption

1. INTRODUCTION

Shamir [15] introduced the first identity-based encryption (IBE) scheme to allow a user to adopt an identity string (*e.g.*, an email address) as a public key in 1984. The major advantage of IBE scheme is to simplify the management of public key certificates. Extensions of IBE have since been widely studied [2, 4, 9, 11, 12]. Hierarchical IBE (HIBE), which reflects an organizational hierarchy, was first presented by Horwitz and Lynn [12]. In HIBE, identities are vectors. A vector of dimension *j* denotes an identity at depth *j* of the hierarchy tree. Many HIBE schemes had been proposed [2, 3, 11, 12]. These schemes possessed two important properties: hierarchical access to messages and decentralized key generation. The property of hierarchical access to messages enables users to decrypt all ciphertexts sent to their subordinates (descendants) in the hierarchy tree. The property of decentralized key generation allows an identity at depth *j* of the hierarchy tree to issue private keys to its child nodes. Essentially, the HIBE is not meant to prevent users from accessing secret messages sent to their subordinates. Thus, it is not suitable for the envi-

Received June 20, 2008; revised December 2, 2008 & March 9, 2009; accepted August 13, 2009.
Communicated by Wen-Guey Tzeng.

* This research was supported by the National Science Council of Taiwan, R.O.C., under the Contract No. NSC 95-2221-E-006-071.

ronment which allows users to protect their privacy from supervision of their ancestor users. In addition, the property of decentralized key generation also may not be suitable in the centralized key escrow[1] environments.

Let us consider the following application environments. An intelligence organization usually has a key generation center (KGC) to generate the private keys for all agents so that the property of centralized key escrow is held. The organization is divided into several sections. Each section has a supervisor who can monitor the communications of his subordinate agents by decrypting the ciphertexts sent to them without knowing their private keys. However, the chief of the organization sometimes requires to send messages to some specified agents wherein the messages cannot be known by their supervisors. Another example is in the representation of any company organized as a hierarchy tree, in which a computer center is fully responsible for the key generation for the company. A new employee obtains his employee identity number and his corresponding private key directly from the computer center. To observe an employee's activity during working hours, a manager is authorized to access encrypted emails sent to his subordinates without knowing their private keys. However, the CEO (chief executive officer) may require to send messages which can only be decrypted by some specified employees rather than all their ancestor users in the hierarchy tree. Obviously, the existing HIBE schemes cannot be directly applied to the above application environments due to the properties of hierarchical access to messages and decentralized key generation.

Some existing identity-based cryptographic schemes, such as identity-based broadcast encryption schemes (ID-based BESs) [10, 14, 16] and identity-based multi-recipient encryption scheme (ID-based MRES) [1], are proposed for the application environments which are similar to the aforementioned ones. In an ID-based BES, a sender encrypts a session key k and sends it to a dynamically changing set of users such that only a privileged subset of users can decrypt it. With the session key k, the subsequent broadcasts can be secured using a conventional private-key cryptosystem, such as DES. The ID-based MRES proposed in [1] allows a sender to encrypt a single message for n specified receivers with low computation complexity. Upon receiving the ciphertext, receiver i, for $i = 1, 2, \ldots, n$, can decrypt it using its own private key. Although both ID-based BESs [10, 14, 16] and the ID-based MRES [1] allow the message sender to encrypt and send a message to a set of specified receivers, they are not especially designed for an organizational hierarchy of users.

This paper aims to present a cryptographic scheme, called identity-based hierarchical designated decryption (IHDD), for the requirements of selectable hierarchical access to messages and centralized key escrow in the hierarchical environment. The selectable hierarchical access to messages is that a user encrypts a message and determines whether the decryption is allowed by (1) only a specified receiver or (2) a specified receiver and all or some of its ancestor users in the hierarchy tree. When a user is allowed to decrypt the ciphertexts sent to its subordinates, it can perform the decryption operation with its own private key without knowing the private keys of its subordinates. The property of centralized key escrow requires all users' private keys generated by the KGC rather than by their ancestor users.

We propose an IHDD scheme based on the bilinear pairing. The proposed scheme

[1] Centralized key escrow is a system in which all cryptographic keys needed to decrypt ciphertexts are held in escrow by a trusted third party. It is used to ensure that the cryptographic keys are controlled and backed up in case they are lost by any user due to a disaster or malicious intent.

can be considered as a combination of Boneh and Boyen's HIBE scheme [2] and Baek *et al.*'s ID-based MRES [1]. However, the purpose and structure of our scheme are different from those of all the previous ones. The proposed IHDD scheme has the following advantages: (1) lower computation complexity: the decryption operation in the proposed scheme needs only one bilinear pairing computation, while the decryption in the existing HIBE schemes and the ID-based MRES [1] needs at least two such computations, and (2) constant length of private keys: each identity's private key in the proposed scheme comprises two elements, and its length is independent of the depth in the hierarchy tree; by contrast, the private key length in existing HIBEs depends on the hierarchy depth.

To show the security of the proposed IHDD scheme, a formal security model is presented using the *selective-identity attack* [2, 5]. The selective-identity (sID) attack, in which the adversary must commit ahead of time to the identity that it intends to attack, is a slightly weaker security notion than the full identity attack [4], in which the adversary is allowed to choose the identity adaptively. The sID attack has been widely adopted for designing a security proof model by the research community. In [7], Chatterjee and Sarkar indicated a technical difficulty in the sID attack, wherein the adversary has to commit to identities even before it knows the set of identities. More precisely, the identity space is usually specified by the setup algorithm of the cryptographic scheme. However, since the actual setup has not been done, there is no real adversary and hence no real target identity. Chatterjee and Sarkar [7] suggested two possible solutions to the difficulty. The first solution is to allow the adversary to commit to binary strings and later when the setup program has been executed, these binary strings are mapped to identities using a collision resistant hash function. Another solution is to run the setup program in two phases. In the first phase, the identity space is specified and is made available to the adversary; then the adversary commits to the identities that it intends to attack; and after obtaining the identities the rest of the setup program is executed.

This paper formally shows that the proposed IHDD scheme is secure against sID adaptive chosen-plaintext attacks (*i.e.*, "sID-CPA-secure") based on the decision bilinear Diffie-Hellman inversion (decision BDHI) assumption [2] in the standard model (*i.e.*, without the use of random oracles). Roughly speaking, the decision BDHI assumption says that no efficient algorithm can distinguish $e(g, g)^{1/x}$ from random, given g, g^x, $g^{(x^2)}$, …, $g^{(x^\kappa)}$ for some positive integer κ. Moreover, a recent result of Canetti *et al.* [6] gives an efficient approach to construct a CCA-secure (*i.e.*, secure against adaptive chosen-ciphertext attacks) public-key encryption scheme from any CPA-secure identity-based encryption scheme. Accordingly, the proposed sID-CPA-secure IHDD scheme can also be converted into an sID-CPA-secure IHDD scheme based on Canetti *et al.*'s approach [6].

The remainder of this paper is organized as follows. Section 2 proposes the definition of security for the IHDD scheme and reviews bilinear groups and the decision BDHI assumption. Section 3 proposes a pairing-based IHDD scheme, and shows its security in the sID attack model without using random oracles. A short conclusion is drawn in section 4.

2. PRELIMINARIES

2.1 sID Secure IHDD Scheme

Let l-IHDD scheme denote an IHDD scheme with the hierarchy tree of maximum depth l. In the l-IHDD scheme, a vector of dimension j represents an identity at depth j of the hierarchy tree, *e.g.* $\mathbf{ID}_j = (I_1, \ldots, I_j)$, $1 \leq j \leq l$. The ancestors of the identity $\mathbf{ID}_j$ are the users whose identities are $\mathbf{ID}_i' = (I_1', \ldots, I_i')$ for $1 \leq i < j$ and $I_a = I_a'$ for $1 \leq a \leq i$. Thus, the identity of an ancestor of the identity $\mathbf{ID}_j$ is said to be the prefix of $\mathbf{ID}_j$. Let S denote the set of all users' identities in the hierarchy tree. Let the symbol *Choice-type* denote that the ciphertext can be decrypted either by (1) only a specified receiver (*i.e.*, *Choice-type* = "*Member-choice*") or (2) a specified receiver and all or some of its ancestor users in the hierarchy tree (*i.e.*, *Choice-type* = "*Member-ancestor-choice*"). Let the symbol *Choice* denote the set of receivers chosen by the message sender, and $Choice \subseteq S$. Therefore, if *Choice-type* = "*Member-choice*", the *Choice* contains only an identity, while if *Choice-type* = "*Member-ancestor-choice*", the *Choice* contains an identity of a specified receiver and the indices of its ancestor users chosen by the message sender.

As in an HIBE scheme [2, 11, 12], the l-IHDD scheme is composed of the following four algorithms: *Setup*, *KeyGen*, *Encrypt*, *Decrypt*.

- **Setup:** On input of the maximum depth l of the hierarchy tree, this algorithm outputs the identity space S, the system public parameters *params* and the master secret keys *master-keys*.
- **KeyGen:** On input of the *master-keys* and a user's identity $\mathbf{ID}_j \in S$, this algorithm outputs the user's private key d_j.
- **Encrypt:** This algorithm takes as input the *params*, *Choice-type*, *Choice*, and a message M. It uses *params* and all identity $\mathbf{ID}^* \in Choice$ to encrypt M and outputs a ciphertext C.
- **Decrypt:** This algorithm takes as input the *Choice-type*, *Choice*, the private key d corresponding to the identity $\mathbf{ID}^* \in Choice$ and a ciphertext C. It outputs a plaintext M.

The following game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ formally defines CCA security of l-IHDD scheme in the sID attack model. This security model can be considered that the adversary wants to decrypt a specific ciphertext sent to a specified receiver and its ancestor users in the hierarchy tree. The adversary can be a registered user and is capable of obtaining the other users private keys except the users in the *Choice*. Additionally, the adversary can collect messages exchanged between users and can ask users to decrypt any messages except the target ciphertext. The security model adopts the solutions suggested by Chatterjee and Sarkar [7]. The setup algorithm is divided into two phases. In the first phase, the identity space S is specified and is made available to the adversary. Then the adversary commits to the identities that it intends to attack. After obtaining the identities, the rest of the setup algorithm is executed. The game is described as follows.

Setup_1: The challenger $\mathcal{C}$ runs the *Setup* algorithm. $\mathcal{C}$ gives $\mathcal{A}$ the identity space S of users.

Commit: The adversary $\mathcal{A}$ outputs a *Choice-type*, a set *Choice* and the identity $\mathbf{ID}^*$ that it tends to attack, where $\mathbf{ID}^* \in Choice$.

Setup_2: The challenger $\mathcal{C}$ continues to run the rest of the *Setup* algorithm. The $\mathcal{C}$ gives $\mathcal{A}$ the resulting system parameters *params* and keeps the *master-keys* to itself.

Phase 1: $\mathcal{A}$ issues queries $q_1, \ldots, q_m$ where query q_i is one of the following

- Private key query $\langle \mathbf{ID}_i \rangle$, where $\mathbf{ID}_i \in S$ and $\mathbf{ID}^* \notin$ *Choice*. $\mathcal{C}$ responds by running algorithm *KeyGen* to generate the private key d_i corresponding to the public key $\mathbf{ID}_i$ and sends d_i to $\mathcal{A}$.
- Decryption query $\langle \mathbf{C}_i \rangle$ for an identity $\mathbf{ID}^* \in$ *Choice*. $\mathcal{C}$ responds by running algorithm *KeyGen* to generate the private key d corresponding to $\mathbf{ID}^*$ (or the relevant prefix as requested). It then runs algorithm *Decrypt* to decrypt the ciphertext $\mathbf{C}_i$ by using the private key d, and sends the resulting plaintext to $\mathcal{A}$.

  $\mathcal{A}$ may query $\mathcal{C}$ adaptively, that is, each query q_i may depend on the replies to $q_1, \ldots, q_{i-1}$.

Challenge: Once $\mathcal{A}$ decides that Phase 1 is over it outputs two equal length plaintexts M_0, $M_1 \in \mathcal{M}$ on which it wishes to be challenged. $\mathcal{C}$ chooses a bit $\bar{b} \in \{0, 1\}$ at random and sets the challenge ciphertext to $\boldsymbol{C}$ = *Encrypt*(*params*, *Choice-type*, *-Choice*, $M_{\bar{b}}$). It sends $\mathbf{C}$ as the challenge to the adversary $\mathcal{A}$.

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

- Private key query $\langle \mathbf{ID}_i \rangle$, where $\mathbf{ID}_i \notin$ *Choice*. The $\mathcal{C}$ responds as in Phase 1.
- Decryption query $\langle \mathbf{C}_i \rangle$, where $\mathbf{C}_i \neq \mathbf{C}$ for an identity $\mathbf{ID}^* \in$ *Choice*. $\mathcal{C}$ responds as in Phase 1.

  These queries may be asked adaptively as in Phase 1.

Guess: Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. $\mathcal{A}$ wins if $b' = \bar{b}$.

The adversary $\mathcal{A}$ is referred as an IND-sID-CCA adversary. The advantage of the adversary $\mathcal{A}$ in attacking the scheme $\mathcal{E}$ is defined as

$$Adv_{E,\mathcal{A}} = \left| \Pr[b' = \bar{b}] - \frac{1}{2} \right|.$$

The probability is over the random bits used by the challenger and the adversary.

Definition 1 An l-IHDD scheme $\mathcal{E}$ is said to be $(t, q_s, q_D, \varepsilon)$-selective identity, adaptive chosen ciphertext secure if for any t-time IND-sID-CCA adversary $\mathcal{A}$ that makes at most q_s chosen private key queries and at most q_D chosen decryption queries we have that $Adv_{\mathcal{E},\mathcal{A}} < \varepsilon$. As shorthand, we say that $\mathcal{E}$ is $(t, q_s, q_D, \varepsilon)$-IND-sID-CCA secure.

Semantic Security Like the description in [2], the selective identity, chosen plaintext security for an l-IHDD scheme is defined as in the previous game, except that the adversary is not allowed to issue any decryption queries. The adversary may still issue adaptive private key queries. This security notion is termed as IND-sID-CPA.

Definition 2 An l-IHDD scheme $\mathcal{E}$ is said to be (t, q_s, ε)-IND-sID-CPA secure if $\mathcal{E}$ is $(t, q_s, 0, \varepsilon)$-IND-sID-CCA secure.

2.2 Bilinear Groups

This subsection briefly reviews the necessary facts about bilinear maps (bilinear pairings) and bilinear map groups. We will follow the notation in [4]:

1. $\mathbb{G}$ and $\mathbb{G}_1$ are two (multiplicative) cyclic groups of prime order p.
2. g is a generator of $\mathbb{G}$.
3. e is a bilinear map e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$.

Let $\mathbb{G}$ and $\mathbb{G}_1$ be two groups as above. A bilinear map is a map e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ with the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1_{\mathbb{G}_1}$.

If the group action in $\mathbb{G}$ can be computed efficiently and there exists a group $\mathbb{G}_1$ and an efficiently computable bilinear map e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ as above, then $\mathbb{G}$ is said to be a bilinear group. Notice that $e(,)$ is symmetric because

$$e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}.$$

2.3 Bilinear Diffie-Hellman (BDH) Assumption

Let $\mathbb{G}$ be a bilinear group of prime order p. The BDH problem [2, 13] in $\mathbb{G}$ is as follows: given a tuple $g, g^a, g^b, g^c \in \mathbb{G}$ as input, output $e(g, g)^{abc} \in \mathbb{G}_1$. An algorithm $\mathcal{A}$ has advantage $\mathcal{F}$ in solving the *decision* BDH problem in $\mathbb{G}$ if

$$\Pr[\mathcal{A}(g, g^a, g^b, g^c) = e(g, g)^{abc}] \geq \varepsilon,$$

where the probability is over the random choice of generator g in $\mathbb{G}^*$, the random choice of a, b, c in $\mathbb{Z}_p$, and the random bits used by $\mathcal{A}$. Similarly, we say that an algorithm $\mathcal{B}$ that outputs $b' \in \{0, 1\}$ has advantage ε in solving the *decision* BDH problem in $\mathbb{G}$ if

$$\left| \Pr[\mathcal{B}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 1] - \Pr[\mathcal{B}(g, g^a, g^b, g^c, T) = 1] \right| \geq \varepsilon,$$

where the probability is over the random choice of generator g in $\mathbb{G}^*$, the random choice of a, b, c in $\mathbb{Z}_p$, the random choice of $T \in \mathbb{G}_1$, and the random bits consumed by $\mathcal{B}$.

Definition 3 We say that the (decision) (t, ε)-*BDH* assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ε in solving the (decision)-*BDH* problem in $\mathbb{G}$.

Occasionally we drop the t and ε and refer to the BDH and Decision BDH assumptions in $\mathbb{G}$.

2.4 Bilinear Diffie-Hellman Inversion (BDHI) Assumption

Let $\mathbb{G}$ be a bilinear group of prime order p, and let g be a generator of $\mathbb{G}$. The κ-BDHI problem in the group $\mathbb{G}$ is defined as follows [2]: given the $(\kappa + 1)$-tuple $(g, -g^x, g^{(x^2)}, \ldots,$

$g^{(x^\kappa)}) \in (\mathbb{G}^*)^{\kappa+1}$ as input, compute $e(g, g)^{1/x} \in \mathbb{G}_1^*$. An algorithm $\mathcal{A}$ has advantage ε in solving κ-BDHI in $\mathbb{G}$ if

$$\Pr[\mathcal{A}(g, g^x, \ldots, g^{(x^\kappa)}) = e(g, g)^{1/x}] \geq \varepsilon,$$

where the probability is over the random choice of x in $\mathbb{Z}_p^*$ and the random bits of $\mathcal{A}$. Similarly, we say that an algorithm $\mathcal{B}$ that outputs $b' \in \{0, 1\}$ has advantage $\mathcal{F}$ in solving the *decision* κ-BDHI problem in $\mathbb{G}$ if

$$\left|\Pr[\mathcal{B}(g, g^x, \ldots, g^{(x^\kappa)}, e(g, g)^{1/x}) = 1] - \Pr[\mathcal{B}(g, g^x, \ldots, g^{(x^\kappa)}, T) = 1]\right| \geq \varepsilon,$$

where the probability is over the random choice of x in $\mathbb{Z}_p^*$, the random choice of $T \in \mathbb{G}_1^*$, and the random bits of $\mathcal{B}$.

Definition 4 We say that the (decision) (t, κ, ε)-*BDHI* assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ε in solving the (decision) κ-*BDHI* problem in $\mathbb{G}$.

For conciseness we sometimes drop the t and ε and simply refer to κ-BDHI and decision κ-BDHI assumptions. It is easy to show that 1-BDHI assumption is equivalent to the standard BDH assumption. It is not known if the κ-BDHI assumption, for $\kappa > 1$ is equivalent to the BDH assumption [2]. Cheon [8] recently proposed an attack to the strong Diffie-Hellman (SDH) related problems. However, if the security of a cryptographic scheme is based on the variant of SDH problems, Cheon also suggests to increase the key size or use a prime p such that both $p + 1$ and $p - 1$ have no small divisor greater than $(\log p)^2$ to avoid the proposed attack in [8].

3. AN IHDD SCHEME BASED ON THE DECISION BDHI ASSUMPTION

3.1 The Proposed Scheme

Setup(l): On input of the maximum depth l of the hierarchy tree, an identity at depth j of the hierarchy tree is a vector of elements in $\mathbb{Z}_p^*$ and denoted as $\mathbf{ID}_j = (I_1, \ldots, I_j) \in (\mathbb{Z}_p^*)^j$, $1 \leq j \leq l$. The set of all users' identities in the hierarchy tree is denoted as S. The symbol $(i_1, i_2, \ldots, i_k) - \mathit{ancestors}_{\mathbf{ID}_j}$ denotes the subset of $\mathbf{ID}_j$'s ancestors in which i_a represents the ancestor at depth a of the hierarchy tree, $1 \leq a \leq k < j$. Let $\mathbb{G}$, $\mathbb{G}_1$, g and e be defined as the above section. To generate system parameters for the l-IHDD scheme, the system (key generation center, KGC) selects different elements $x, y_1, y_2, \ldots, y_l, z \in \mathbb{Z}_p^*$ at random and defines $X = g^x$, $Z = g^z$ and $Y_i = g^{y_i}$, $i = 1, 2, \ldots, l$. The public parameters *params* and the secret keys *master-keys* are given by

$$\mathit{params} = (g, g^x, g^{y_1}, g^{y_2}, \ldots, g^{y_l}, g^z) = (S, g, X, Y_1, Y_2, \ldots, Y_l, Z),$$
$$\mathit{master\text{-}keys} = (x, y_1, y_2, \ldots, y_l, z).$$

***KeyGen*(*master-keys*, $\mathbf{ID}_j$):** To generate the private key d_j for an identity $\mathbf{ID}_j \in S$ and $\mathbf{ID}_j = (I_1, \ldots, I_j) \in (\mathbb{Z}_p^*)^j$ of depth $j \leq l$, the KGC chooses a number $r_j \in \mathbb{Z}_p^*$ at random and calculates

$$K_j = g^{\frac{1}{x+r_j(I_1y_1+I_2y_2+\cdots+I_jy_j+j\cdot z)}} \in \mathbb{G},$$

where $x + r_j(I_1y_1 + I_2y_2 + \ldots + I_jy_j + j \cdot z) \neq 0 \pmod p$. The KGC outputs (r_j, K_j) as the private key d_j for $\mathbf{ID}_j$. Notice that d_j must be sent to $\mathbf{ID}_j$ using a secure communication channel.

***Encrypt*(*params*, *Choice-type*, *Choice*, *M*):** To encrypt a message $M \in \mathbb{G}_1$ under the *Choice-type* and the set *Choice*,

1. If *Choice-type* = *Member-choice*, *Choice* = $\{\mathbf{ID}_j\}$ where $\mathbf{ID}_j \in S$ and $\mathbf{ID}_j = (I_1, I_2, \ldots, I_j) \in (\mathbb{Z}_p^*)^j$, then the sender chooses a number $s \in \mathbb{Z}_p^*$ at random and outputs a ciphertext

$$\mathbf{C} = (X^s, (Y_1^{I_1} \cdot Y_2^{I_2} \cdot \cdots \cdot Y_j^{I_j} \cdot Z^j)^s, e(g,g)^s \cdot M) \in \mathbb{G}^* \times \mathbb{G}^* \times \mathbb{G}_1^*.$$

2. If *Choice-type* = *Member-ancestor-choice* and *Choice* = $\{\mathbf{ID}_j\} \cup (i_1, i_2, \ldots, i_k)$ – *ancestors*$_{\mathbf{ID}_j}$ where $\mathbf{ID}_j \in S$ and $\mathbf{ID}_j = (I_1, \ldots, I_j) \in (\mathbb{Z}_p^*)^j$ and $1 \leq i_1 < i_2 < \ldots < i_k < j$, then the sender chooses a number $s \in \mathbb{Z}_p^*$ at random and outputs a ciphertext

$$\mathbf{C} = (X^s, (Y_1^{I_1} Y_2^{I_2} \cdots Y_{i_1}^{I_{i_1}} \cdot Z^{i_1})^s, (Y_{i_1+1}^{I_{i_1+1}} \cdots Y_{i_2}^{I_{i_2}} \cdot Z^{i_2-i_1})^s, \cdots, (Y_{i_k+1}^{I_{i_k+1}} \cdots Y_{i_{k+1}}^{I_{i_{k+1}}} \cdot Z^{i_{k+1}-i_k})^s,$$
$$e(g,g)^s \cdot M) \in (\mathbb{G}^*)^{k+2} \times \mathbb{G}_1^*$$

where $i_{k+1} = j$. (To simplify the expression, we use i_{k+1} to represent j, *i.e.*, $\mathbf{ID}_{i_{k+1}} = \mathbf{ID}_j$)

Notice that the $e(g, g)$ can be precomputed once and for all so that encryption does not require any pairing computations.

***Decrypt*(*Choice-type*, *Choice*, *C*):** To decrypt a ciphertext $\mathbf{C}$ encrypted under the *Choice-type* and the set *Choice*,

1. If *Choice-type* = *Member-choice*, *Choice* = $\{\mathbf{ID}_j\}$ where $\mathbf{ID}_j \in S$ and $\mathbf{ID}_j = (I_1, \ldots, I_j) \in (\mathbb{Z}_p^*)^j$ and $\mathbf{C} = (A, B, C)$, the receiver $\mathbf{ID}_j$ uses the private key $d = (r_j, K_j)$ to output

$M = C/e(A \cdot B^{rj}, K_j) \in \mathbb{G}_1$.

Indeed, for a valid ciphertext, we have

$$\frac{C}{e(A \cdot B^{r_j}, K_j)} = \frac{e(g,g)^s \cdot M}{e(g^{xs+(I_1y_1+I_2y_2+\cdots+I_jy_j+jz)s\cdot r_j}, K_j)} = \frac{e(g,g)^s \cdot M}{e(g,g)^s} = M.$$

2. If *Choice-type* = *Member-ancestor-choice* and *Choice* = $\{\mathbf{ID}_j\} \cup (i_1, i_2, \ldots, i_k)$ – *ancestors*$_{\mathbf{ID}_j}$ where $\mathbf{ID}_j \in S$ and $\mathbf{ID}_j = (I_1, \ldots, I_j) \in (\mathbb{Z}_p^*)^j$ and $\mathbf{C} = (A, B_1, B_2, \ldots, B_{k+1}, C)$, the

receiver $\mathbf{ID}_{i_\tau}$ uses the private key $d = (r_{i_\tau}, K_{i_\tau})$ to output

$$M = C/e(A\cdot(B_1 \cdot B_2 \cdot \ \dots\ \cdot B_\tau)^{r_{i_\tau}}, K_{i_\tau}) \in \mathbb{G}_1$$

where $\mathbf{ID}_{i_\tau} \in \mathit{Choice}$, $\mathbf{ID}_{i_\tau} = (I_1, \dots, I_{i_\tau})$, $1 \le \tau \le k + 1$ and $i_{k+1} = j$. Indeed, for a valid ciphertext, we have

$$\frac{C}{e(A\cdot(B_1 \cdot B_2 \cdot \dots \cdot B_\tau)^{r_{i_\tau}}, K_{i_\tau})} = \frac{e(g,g)^s \cdot M}{e(g^{xs+(I_1y_1+I_2y_2+\dots+I_{i_\tau}y_{i_\tau}+i_\tau z)s\cdot r_{i_\tau}}, K_{i_\tau})} = \frac{e(g,g)^s \cdot M}{e(g,g)^s} = M.$$

Performance In the proposed l-IHDD scheme, the ciphertext given to an identity at level j can also be decrypted by some of its ancestor identities selected by the sender. In terms of efficiency, the decryption in the proposed l-IHDD scheme requires only one pairing computation, as opposed to at least two pairing computations in the previous HIBE schemes [2, 3] and the ID-based MRES [1]. Since $e(g, g)$ can be precomputed and known by all identities, the proposed l-IHDD scheme and the above related works do not require any pairing computation during the encryption operation. Although in the l-IHDD scheme the length of the ciphertext is varied depending on the number of the receivers, the private key of each user consists of only two elements and the length is independent of the hierarchy depth, whereas in previous HIBE schemes the length of the private keys is varied depending on the hierarchy depth. We summarize the above comparison in Table 1.

Table 1. The comparison between the proposed *l*-IHDD scheme and some related encryptions.

	HIBE in [2]	HIBE in [3]	ID-based MRES in [1]	Our l-IHDD scheme
pairing required by decryption	$j + 1$	2	2	1
pairing required by encryption	0	0	0	0
the length of the private key	varied	varied	constant size	constant size
the length of the ciphertext	varied	constant size	varied	varied
selectable hierarchical access	no	no	yes	yes
centralized key escrow	no	no	yes	yes

j denotes the depth of the specified receiver in the hierarchy tree.

3.2 Security Analysis

By adapting the proof techniques proposed in [2], the proposed l-IHDD scheme is first shown to be selective identity, chosen plaintext (IND-sID-CPA) secure under the decision κ-BDHI assumption without using random oracles. The descriptions that how to achieve selective identity, chosen ciphertext security (IND-sID-CCA) and how to handle arbitrary identities are then given.

Theorem 1 Suppose the (t, κ, ε)-decision BDHI assumption holds in $\mathbb{G}$. Then the proposed l-IHDD scheme is (t', q_s, ε')-selective identity, chosen plaintext (IND – sID – CPA) secure for any depth l, any $q_s < \kappa$, $\varepsilon' \le \varepsilon + \frac{q_s}{p}$ and $t' \le t - \Theta((l + q_s)t'')$, where t'' is the maximum time for an exponentiation in $\mathbb{G}$.

Proof: Suppose that $\mathcal{A}$ makes q_s private key queries and has advantage ε' in attacking the proposed l-IHDD scheme. Algorithm $\mathcal{B}$ will use $\mathcal{A}$ to solve the decision κ-BDHI problem in $\mathbb{G}$.

Given $(g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^{\kappa})}, T) \in (\mathbb{G}^*)^{\kappa+1} \times \mathbb{G}_1^*$ for some unknown $\alpha \in \mathbb{Z}_p^*$, with a game, $\mathcal{B}$ will determine whether T is equal to $e(g, g)^{1/\alpha}$ or not. The output is 1 if $T = e(g, g)^{1/\alpha}$, otherwise the output is 0. $\mathcal{B}$ interacts with $\mathcal{A}$ in a game as follows.

Preparation Algorithm $\mathcal{B}$ produces a generator $h \in \mathbb{G}^*$ and $\kappa - 1$ pairs of the form $(w_i, h^{1/(\alpha+w_i)})$ for different random numbers $w_1, w_2, \ldots, w_{\kappa-1} \in \mathbb{Z}_p^*$ as following steps:

1. Choose random numbers $w_1, w_2, \ldots, w_{\kappa-1} \in \mathbb{Z}_p^*$ and let $f(z) = \prod_{i=1}^{\kappa-1}(z + w_i)$ be a polynomial function of degree $\kappa - 1$. Expand the terms of f into $f(z) = \sum_{i=0}^{\kappa-1} c_i z^i$. The constant term c_0 is non-zero.
2. Use values $(g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^{\kappa})})$ to compute

$$h = \prod_{i=0}^{\kappa-1} (g^{(\alpha^i)})^{c_i} = g^{f(\alpha)} \text{ and } u = \prod_{i=1}^{\kappa} (g^{(\alpha^i)})^{c_{i-1}} = g^{\alpha f(\alpha)} = h^{\alpha}.$$

3. Check that $h \in \mathbb{G}^*$. If $h = 1_{\mathbb{G}}$ in $\mathbb{G}$, it would mean that $w_i = -\alpha$ for some w_i. Then $\mathcal{B}$ would be able to solve the challenge directly. We thus assume that all $w_i \ne -\alpha$.
4. To construct the pairs $(w_i, h^{1/(\alpha+w_i)})$ for $i = 1, 2, \ldots, \kappa - 1$, $\mathcal{B}$ takes $f_i(z) = f(z)/(z + w_i)$ and calculates

$$h^{1/(\alpha+w_i)} = g^{f_i(\alpha)} = \prod_{i=0}^{\kappa-2} (g^{(\alpha^i)})^{\hat{c}_i},$$

   where $\hat{c}_i$ is the coefficients of the polynomial $f_i(z)$.
5. Let $f(z) + c_0 = \sum_{i=0}^{\kappa-1} c_i' z^i$, where $c_0' = 2c_0$ and $c_i' = c_i$ for $i = 1, 2, \ldots, \kappa - 1$. $\mathcal{B}$ computes

$$T_h = T^{(c_0^2)} \cdot T_0,$$

   where $T_0 = \prod_{i=0}^{\kappa-1} \prod_{j=0}^{\kappa-2} e(g^{(\alpha^i)}, g^{(\alpha^j)})^{c_i' c_{j+1}} = e(g^{f(\alpha)+c_0}, g^{[f(\alpha)-c_0]/\alpha})$. Observe that if $T = e(g, g)^{1/\alpha}$, then $T_h = e(g^{f(\alpha)/\alpha}, g^{f(\alpha)}) = e(h, h)^{1/\alpha}$.

   The values h, u, T_h and the pairs $(w_i, h^{1/(\alpha+w_i)})$ for $i = 1, 2, \ldots, \kappa - 1$ will be used throughout the simulation.

Setup 1: On input of the maximum depth l of the hierarchy tree, an identity at depth j of the hierarchy tree is a vector of elements in $\mathbb{Z}_p^*$ and denoted as $\mathbf{ID}_j = (I_1, \ldots, I_j) \in (\mathbb{Z}_p^*)^j$, $1 \le j \le l$. $\mathcal{B}$ gives $\mathcal{A}$ the set S of all users' identities in the hierarchy tree.

Commit: $\mathcal{A}$ outputs a *Choice-type*, a set *Choice* and an identity $\mathbf{ID}^*$, where $\mathbf{ID}^* \in S$ and $\mathbf{ID}^* = (I_1^*, I_2^*, \ldots, I_n^*) \in (\mathbb{Z}_p^*)^n \cap$ *Choice* of depth $n \leq l$ that it tends to attack.

Setup 2: $\mathcal{B}$ appends $l - n$ unit elements to $\mathbf{ID}^*$ such that $\mathbf{ID}^* = (I_1^*, I_2^*, \ldots, I_n^*, 1, 1, \ldots, 1)$ is a vector of length l if necessary. The $\mathbf{ID}^*$ can be denoted as $\mathbf{ID}^* = (I_1^*, I_2^*, \ldots, I_l^*)$. $\mathcal{B}$ generates the system parameters *params* $= (h, X, Y_1, \ldots, Y_l, Z)$ as follows.

Case 1: If *Choice-type* = *Member-choice* and *Choice* = $\{\mathbf{ID}^*\}$, then $\mathcal{B}$ runs the following steps:

1. Choose different numbers $a_1, a_2, \ldots, a_l, b \in \mathbb{Z}_p^*$ at random.
2. Compute $X = u = h^\alpha$, $Y_i = h^{a_i}$ for any $i = 1, 2, \ldots, l$, and $Z = (u^b h^{-(a_1 I_1^* + a_2 I_2^* + \cdots + a_n I_n^*)})^{n^{-1}}$. We implicitly define the master keys $x = \alpha$, $y_i = a_i$ and $z = n^{-1} \cdot [b\alpha - (a_1 I_1^* + a_2 I_2^* + \ldots + a_n I_n^*)]$ so that $X = h^x$, $Y_i = h^{y_i}$, and $Z = h^z$.
3. Publish the public parameters *params* $= (h, X, Y_1, Y_2, \ldots, Y_l, Z)$. Note that $X, Y_1, Y_2, \ldots, Y_l, Z$ are independent of $\mathbf{ID}^*$ in the adversary's view.
4. Let an integer β be used to count the pairs $(w_i, h^{1/(\alpha+w_i)})$ which had been used in the simulation. Set the initial value $\beta = 0$.

Case 2: If *Choice-type* = *Member-ancestor-choice* and *Choice* = $\{\mathbf{ID}_j\} \cup (i_1, i_2, \ldots, i_k) -$ *ancestors*$_{\mathbf{ID}_j}$, then $\mathcal{B}$ runs the following steps:

1. Choose different numbers $a_1, a_2, \ldots, a_l, b \in \mathbb{Z}_p^*$ at random.
2. Compute $X = u = h^\alpha$, $Z = h^b$, $Y_i = h^{a_i}$ for any $i \in \{1, 2, \ldots, l\} \backslash \{i_1, i_2, \ldots, i_k, n\}$, and $Y_{i_\tau} = (u^{a_\tau} h^{-(i_\tau - i_{\tau-1})b - \sum_{j=i_{\tau-1}+1}^{i_\tau - 1} I_j^* a_j})^{(I_{i_\tau}^*)^{-1}}$ for any $i_\tau \in \{i_1, i_2, \ldots, i_k, n\}$. We implicitly define the master keys $x = \alpha$, $z = b$, $y_i = a_i$ and $y_{i_\tau} = (I_{i_\tau}^*)^{-1} \cdot [a_{i_\tau}\alpha - (i_\tau - i_{\tau-1})b - \sum_{j=i_{\tau-1}+1}^{i_\tau - 1} I_j^* a_j]$ for any $i \in \{1, 2, \ldots, l\} \backslash \{i_1, i_2, \ldots, i_k, n\}$ and $i_\tau \in \{i_1, i_2, \ldots, i_k, n\}$ so that $X = h^x$, $Y_i = h^{y_i}$, and $Z = h^z$.
3. Publish the public parameters *params* $= (h, X, Y_1, Y_2, \ldots, Y_l, Z)$. Note that $X, Y_1, Y_2, \ldots, Y_l, Z$ are independent of $\mathbf{ID}^*$ in the adversary's view.
4. Let an integer β be used to count the pairs $(w_i, h^{1/(\alpha+w_i)})$ which had been used in the simulation. Set the initial value $\beta = 0$.

Phase 1: $\mathcal{A}$ requests at most q_s, $q_s < \kappa$, private key queries.

Case 1: If *Choice-type* = *Member-choice*, $\mathcal{B}$ considers a private key query corresponding to its $\mathbf{ID}_\lambda = (I_1, \ldots, I_\lambda) \in (\mathbb{Z}_p^*)^\lambda$, where $\lambda \leq l$, $\mathbf{ID}_\lambda \in S$ and $\mathbf{ID}_\lambda \neq \mathbf{ID}^*$. Algorithm $\mathcal{B}$ responds the private key query by performing the following steps:

1. Set $\beta = \beta + 1$.
2. Let $(w_\beta, h^{1/(\alpha+w_\beta)})$ be the βth pair produced in the preparation step. Define $h_\beta = h^{1/(\alpha+w_\beta)}$.
3. $\mathcal{B}$ derives a private key for the identity $\mathbf{ID}_\lambda = (I_1, I_2, \ldots, I_\lambda)$ as follows:
   (a) Construct an $r_\lambda \in (\mathbb{Z}_p^*)$ satisfying the equation

$$(1+r_\lambda\cdot\lambda n^{-1}b)(\alpha+w_\beta)=x+r_\lambda\cdot\left(\lambda\cdot z+\sum_{i=1}^{\lambda}I_iy_i\right).$$

Let $x = \alpha$, $y_i = a_i$ and $z = n^{-1}\cdot[b\alpha - (a_1I_1^* + a_2I_2^* + \ldots + a_nI_n^*)]$, we can get

$$r_\lambda=\frac{w_\beta}{\sum_{i=1}^{\lambda}I_ia_i-n^{-1}\lambda\cdot\sum_{i=1}^{n}I_i^*a_i-n^{-1}\lambda bw_\beta}\in\mathbb{Z}_p^*.$$

(b) If $r_\lambda = -1/(n^{-1}\lambda b)$, it implies $x+r_\lambda\cdot(\lambda\cdot z+\sum_{i=1}^{\lambda}I_iy_i)=0$. $\mathcal{B}$ returns failure and aborts this game. (This incident is denoted as a Failure event.)

(c) Otherwise, compute

$$h_\beta^{1/(1+r_\lambda\cdot\lambda n^{-1}b)}=h^{1/[x+r_\lambda\cdot(\lambda\cdot z+\sum_{i=1}^{\lambda}I_iy_i)]}.$$

Since w_β is uniform in $\mathbb{Z}_p^*\backslash\{-\alpha\}$ and is currently independent of $\mathcal{A}$'s view, then r_λ is uniformly distributed among all elements in $\mathbb{Z}_p^*$ for which

$$x+r_\lambda\cdot\left(\lambda z+\sum_{i=1}^{\lambda}I_iy_i\right)\neq 0 \text{ and } r_\lambda\neq -1/(n^{-1}\lambda b).$$

Therefore, the private key is $(r_\lambda, h_\beta^{1/(1+r_\lambda\cdot\lambda n^{-1}b)})$ for $\mathbf{ID}_\lambda = (I_1, \ldots, I_\lambda)$. $\mathcal{B}$ gives $\mathcal{A}$ the private key.

Case 2: If *Choice-type* = *Member-ancestor-choice*, $\mathcal{B}$ considers a private key query corresponding to its $\mathbf{ID}_\lambda = (I_1, I_2, \ldots, I_\lambda) \in (\mathbb{Z}_p^*)^\lambda$ where $\lambda \leq l$ and $\mathbf{ID}_\lambda \notin$ *Choice* $(= \{\mathbf{ID}^*\} \cup (i_1, i_2, \ldots, i_k) - \mathit{ancestors}_{\mathbf{ID}^*})$. Before $\mathcal{B}$ runs the following steps, $\mathcal{B}$ appends $l - \lambda$ elements to $\mathbf{ID}_\lambda$ such that $\mathbf{ID}_\lambda = (I_1, I_2, \ldots, I_\lambda, 0, \ldots, 0)$ is a vector of length l. Algorithm $\mathcal{B}$ responds the private key query by running:

1. Set $\beta = \beta + 1$.
2. Let $(w_\beta, h^{1/(\alpha+w_\beta)})$ be the βth pair produced in the preparation step. Define $h_\beta = h^{1/(\alpha+w_\beta)}$.
3. $\mathcal{B}$ derives a private key for the identity $\mathbf{ID}_\lambda = (I_1, I_2, \ldots, I_\lambda, 0, \ldots, 0)$ as follows.
   (a) Construct an $r_\lambda \in \mathbb{Z}_p^*$ satisfying the following equation

$$\left(1+r_\lambda\cdot\sum_{i_\tau\in\{i_1,i_2,\cdots,i_k,n\}}I_{i_\tau}(I_{i_\tau}^*)^{-1}a_{i_\tau}\right)(\alpha+w_\beta)=x+r_\lambda\cdot\left(\lambda\cdot z+\sum_{i=1}^{\lambda}I_iy_i\right)$$
$$=x+r_\lambda\cdot\left(\lambda\cdot z+\sum_{i=1}^{l}I_iy_i\right).$$

Let $x = \alpha$, $z = b$, $y_i = a_i$ for any $i \in \{1, 2, \ldots, l\}\backslash\{i_1, i_2, \ldots, i_k, n\}$, and $y_{i_\tau} = (I_{i_\tau}^*)^{-1}\cdot[a_{i_\tau}\alpha-(i_\tau-i_{\tau-1})b-\sum_{j=i_{\tau-1}+1}^{i_\tau-1}I_j^*a_j]$ for any $i \in \{i_1, i_2, \ldots, i_k, n\}$, we can get

$$r_\lambda = \frac{w_\beta}{\lambda b + \sum\limits_{i \in \{1,\cdots,l\} \setminus \{i_1,\cdots,i_k,n\}} I_i a_i - \Delta},$$

where $\Delta = \sum\limits_{i_\tau \in \{i_1,\cdots,i_k,n\}} I_{i_\tau} (I^*_{i_\tau})^{-1} \left(w_\beta a_{i_\tau} + (i_\tau - i_{\tau-1}) b + \sum\limits_{j=i_{\tau-1}+1}^{i_\tau - 1} I^*_j a_j \right).$

(b) If $r_\lambda = -1/\sum_{i_\tau \in \{i_1,i_2,\cdots,i_k,n\}} (I_{i_\tau} (I^*_{i_\tau})^{-1} a_{i_\tau})$, it implies

$$x + r_\lambda \cdot (\lambda \cdot z + \sum\nolimits_{i=1}^{\lambda} I_i y_i) = 0.$$

$\mathcal{B}$ returns failure and aborts this game. (This incident is denoted as a Failure event.)

(c) Otherwise, compute

$$h_\beta^{1/(1+r_\lambda \cdot \sum_{i_\tau \in \{i_1,i_2,\cdots,i_k,n\}} I_{i_\tau} (I^*_{i_\tau})^{-1} a_{i_\tau})} = h^{1/[x + r_\lambda \cdot (\lambda \cdot z + \sum_{i=1}^{\lambda} I_i y_i)]}.$$

Since w_β is uniform in $\mathbb{Z}_p^* \setminus \{-\alpha\}$ and is currently independent of $\mathcal{A}$'s view, then r_λ is uniformly distributed among all elements in $\mathbb{Z}_p^*$ for which

$$x + r_\lambda \cdot \left(\lambda z + \sum_{i=1}^{\lambda} I_i y_i \right) \neq 0 \text{ and } r_\lambda \neq -1/ \sum_{i_\tau \in \{i_1,i_2,\cdots,i_k,n\}} I_{i_\tau} (I^*_{i_\tau})^{-1} a_{i_\tau}.$$

Therefore, the private key is $(r_\lambda, h_\beta^{1/(1+r_\lambda \cdot \sum_{i_\tau \in \{i_1,i_2,\cdots,i_k,n\}} I_{i_\tau} (I^*_{i_\tau})^{-1} a_{i_\tau})})$ for $\mathbf{ID}_\lambda = (I_1, I_2, \ldots, I_\lambda)$. $\mathcal{B}$ gives $\mathcal{A}$ the private key.

Challenge: $\mathcal{A}$ outputs two messages $M_0, M_1 \in \mathbb{G}_1$. $\mathcal{B}$ chooses a bit $\bar{b} \in \{0, 1\}$ and a number $\rho \in \mathbb{Z}_p^*$ at random.

1. If *Choice-type* = *Member-choice*, then it responds with the ciphertext $\mathbf{C} = (h^\rho, h^{b\rho}, T_h^\rho \cdot M_{\bar{b}})$. Define $s = \rho/\alpha$. On the one hand, if $T_h = e(h, h)^{1/\alpha}$, then we have

$$h^\rho = h^{\alpha s} = X^s,$$
$$h^{b\rho} = (h^{b\alpha})^s = (h^{I_1^* y_1 + I_2^* y_2 + \cdots + I_n^* y_n + nz})^s = (Y_1^{I_1^*} \cdots Y_n^{I_n^*} Z^n)^s,$$
$$T_h^\rho = e(h, h)^{\rho/\alpha} = e(h, h)^s.$$

2. If *Choice-type* = *Member-ancestor-choice*, it responds with the ciphertext

$$\mathbf{C} = (h^\rho, h^{a_{i_1}\rho}, h^{a_{i_2}\rho}, \cdots, h^{a_{i_{k+1}}\rho}, T_h^\rho \cdot M_{\bar{b}}),$$

where $i_{k+1} = n$. Define $s = \rho/\alpha$. On the one hand, if $T_h = e(h, h)^{1/\alpha}$, then we have

$$h^{\rho} = h^{\alpha s} = X^{s},$$

$$h^{a_{i_\tau}\rho} = (h^{a_{i_\tau}\alpha})^{s} = (h^{(i_\tau - i_{\tau-1})z + \sum_{j=i_{\tau-1}+1}^{i_\tau} I_j^* y_j})^{s} = (Y_{i_{\tau-1}+1}^{I_{i_{\tau-1}+1}^*} \ldots Y_{i_\tau}^{I_{i_\tau}^*} Z^{i_\tau - i_{\tau-1}})^{s}, i_0 = 1, 1 \le \tau \le k+1,$$

$$T_h^{\rho} = e(h, h)^{\rho/\alpha} = e(h, h)^{s}.$$

It follows that **C** is a valid encryption of $M_{\bar{b}}$ under $\mathbf{ID}^* = (I_1^*, I_2^*, \ldots, I_n^*)$, with the uniformly distributed randomization value $s = \rho/\alpha \in \mathbb{Z}_p^*$. On the other hand, when T_h is uniform in $\mathbb{G}_1 \backslash \{T_0\}$, then **C** is independent of the bit $\bar{b}$ in the adversary's view.

Phase 2: $\mathcal{A}$ requests more private key queries, for a total of at most $q_s < \kappa$. Algorithm $\mathcal{B}$ responds as that in Phase 1.

Guess: Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = \bar{b}$, then $\mathcal{B}$ outputs 1 meaning $T = e(g, g)^{1/\alpha}$. Otherwise, it outputs 0 meaning $T \neq e(g, g)^{1/\alpha}$.

When the input T satisfies $T = e(g, g)^{1/\alpha}$, it implies $T_h = e(h, h)^{1/\alpha}$ in which case $\mathcal{A}$ has the probability $|\Pr[b' = \bar{b}] - 1/2| > \varepsilon'$. On the other hand, when T is uniform and independent in $\mathbb{G}_1^*$, T_h is uniform and independent in $\mathbb{G}_1 \backslash \{T_0\}$ in which case $\Pr[b' = \bar{b}] = 1/2$. Since the probability that $\mathcal{A}$ outputs b' with $b' = \bar{b}$ is at least $\frac{1}{2} + \varepsilon'$ and the Failure event happens with the probability at most $\frac{q_s}{p}$, the probability that $\mathcal{B}$ outputs 1 in case of $T = e(g, g)^{1/\alpha}$ is at least $\frac{1}{2} + \varepsilon' - \frac{q_s}{p}$. Therefore, when $x = \alpha$ is uniform in $\mathbb{Z}_p^*$ and T is uniform in $\mathbb{G}_1^*$ we have that

$$\begin{aligned}\varepsilon &= \left|\Pr[\mathcal{B}(g, g^{\alpha}, g^{(\alpha^2)}, \cdots, g^{(\alpha^{\kappa})}, e(g, g)^{1/\alpha}) = 1] - \Pr[\mathcal{B}(g, g^{\alpha}, g^{(\alpha^2)}, \cdots, g^{(\alpha^{\kappa})}, T) = 1]\right| \\ &\geq \left|\left(\frac{1}{2} + \varepsilon' - \frac{q_s}{p}\right) - \frac{1}{2}\right| = \varepsilon' - \frac{q_s}{p}\end{aligned}$$

as required. This completes the proof of the theorem. ❑

3.3 The Construction of sID-CCA-secure IHDD Scheme

Canetti *et al.* [6] recently presented an efficient approach to construct a CCA-secure public-key encryption scheme from an IBE scheme. Their approach is briefly described as follows [6]. The public key of the new scheme is the master public key *PK* of the IBE scheme and the secret key is the corresponding master secret key. To encrypt a message with respect to a public key *PK*, the sender first generates a key-pair (*vk*, *sk*) for a strong one-time signature scheme[2], and then encrypts the message with respect to the "identity" *vk*. The resulting ciphertext **C** is then signed using *sk* to obtain a signature σ. The final ciphertext consists of the verification key *vk*, the IBE ciphertext **C** and the signature σ. To decrypt a ciphertext (*vk*, **C**, σ), the receiver first verifies the signature on **C** with respect to *vk* (and outputs **invalid** if the verification fails). The receiver then derives the secret key SK_{vk} corresponding to the "identity *vk*, and uses SK_{vk} to decrypt the ciphertext **C** as per the

[2] A "strong" signature scheme has the property that it is infeasible to create even a different signature on the same message.

underlying IBE scheme. Canetti *et al.* [5, 6] extended the above approach to obtain a simple conversion from any CPA-secure binary tree encryption (BTE) scheme to a CCA-secure HIBE. Accordingly, the proposed l-IHDD scheme can also be extended to an IND-sID-CCA secure l-IHDD scheme using Canetti *et al.*'s approach [6]. Furthermore, we can extend the proposed l-IHDD scheme to handle arbitrary identities $\mathbf{ID}_l = (I_1, I_2, \ldots, I_l)$ with $I_j \in \{0, 1\}^*$ (as opposed to $I_j \in (\mathbb{Z}_p^*)$) by first hashing each I_j using a collision resistant hash function H: $\{0, 1\}^* \to \mathbb{Z}_p^*$ in the key generation and encryption algorithms. A standard argument shows that if the original l-IHDD scheme is IND-sID-CCA secure, then so is the l-IHDD scheme with the additional collision resistant hash function.

4. CONCLUSION

This paper has proposed a pairing-based l-IHDD scheme. The proposed scheme allows an encrypted message to be decrypted either by a specified identity or by a specified identity and all or some of its ancestor identities in the hierarchy tree. The private key of an identity contains only two elements that are generated by a KGC, rather than by its parent identity. The proposed l-IHDD scheme has low computation complexity in which the decryption operation needs only one bilinear pairing computation and the encryption operation needs no pairing computation. We have shown that the proposed scheme is IND-sID-CPA secure in the standard model under decision BDHI assumption. In addition, the IND-sID-CPA secure l-IHDD scheme can be converted into an IND-sID-CCA secure l-IHDD scheme based on the construction method proposed by Canetti *et al.* [6]. However, how to construct an l-IHDD scheme based on other security assumptions, and even with a lower computation complexity than the proposed one, could be an interesting topic of future research. In addition, although the private key of each user consists of only two elements and the length is independent of the hierarchy depth in the proposed scheme, the length of the ciphertext is varied depending on the number of the receivers. Thus, how to construct an l-IHDD scheme with constant size ciphertext would also be an interesting topic of future research.

ACKNOLEDGMENT

The authors would like to thank the editor and the anonymous reviewers for their very helpful and valuable comments to enhance the clarity of the manuscript.

REFERENCES

1. J. Baek, R. Safavi-Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in *Proceedings of Public Key Cryptography*, LNCS 3386, 2005, pp. 380-397.
2. D. Boneh and X. Boyen, "Efficient selective-ID identity based encryption without random oracles," in *Proceedings of Advances in Cryptology – Eurocrypt*, LNCS 3027, 2004, pp. 223-238.
3. D. Boneh, X. Boyen, and E. Goh, "Hierarchical identity based encryption with con-

stant size ciphertext," in *Proceedings of Advances in Cryptology – Eurocrypt*, LNCS 3494, 2005, pp. 440-456.

4. D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of Advances in Cryptology – Crypto*, LNCS 2139, 2001, pp. 213-229.
5. R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," in *Proceedings of Advances in Cryptology – Eurocrypt*, LNCS 2656, 2003, pp. 255-271.
6. R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Proceedings of Advances in cryptology – Eurocrypt*, LNCS 3027, 2004, pp. 207-222.
7. S. Chatterjee and P. Sarkar, "Generalization of the selective-id security model for HIBE protocols," in *Proceedings of Public Key Cryptography*, LNCS 3958, 2006, pp. 241-256.
8. J. H. Cheon, "Security analysis of the strong Diffie-Hellman problem," in *Proceedings of Advances in Cryptology – Eurocrypt*, LNCS 4004, 2006, pp. 1-11.
9. C. Cocks, "An identity based encryption based on quadratic residues," in *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, LNCS 2260, 2002, pp. 360-363.
10. X. Du, Y. Wang, J. Ge, and Y. Wang, "An ID-based broadcast encryption scheme for key distribution," *IEEE Transactions on Broadcasting*, Vol. 51, 2005, pp. 264-266.
11. C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Proceedings of Advances in Cryptology*, LNCS 2501, 2002, pp. 548-566.
12. J. Horwitz and B. Lynn, "Towards hierarchical identity-based encryption," in *Proceedings of Advances in Cryptology – Eurocrypt*, LNCS 2332, 2002, pp. 466-481.
13. A. Joux, "A one round protocol for tripartite Diffiie-Hellman," in *Proceedings of the 4th International Symposium on Algorithmic Number Theory*, LNCS 1838, 2000, pp. 385-394.
14. J. W. Lee, Y. H. Hwang, and P. J. Lee, "Efficient pubic key broadcast encryption using identifier of receivers," in *Proceedings of Information Security Practice and Experience: Second International Conference*, LNCS 3909, 2006, pp. 153-164.
15. A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of Advances in Cryptolog*, LNCS 196, 1984, pp. 47-53.
16. C. Yang, X. Cheng, W. Ma, and X. Wang, "A new ID-based broadcast encryption scheme," in *Proceedings of Autonomic and Trusted Computing*, LNCS 4158, 2006, pp. 487-492.

Shu-Hui Chang (張淑慧) received her B.S. degree in Applied Mathematics from National Cheng Kung University, Taiwan, in 1984, and her M.S. degree in Applied Mathematics from National Cheng Kung University, Taiwan, in 1988. She works as an instructor in Center for General Education, Southern Taiwan University. Her research interests include cryptography, information security.

Chuan-Ming Li (李泉明) received the M.S. and Ph.D. degrees in Computer Science from National Cheng Kung University in 1994 and 2008, respectively. He is currently an assistant professor in the Department of Information Management, Shu-Zen College of Medicine and Management, Kaohsiung, Taiwan. His research interests focus on data security and cryptography. More recently his research interests include the study of quantum cryptography.

Tzonelih Hwang (黃宗立) received the undergraduate degree from National Cheng Kung University in 1980, and the M.S. and Ph.D. degrees in computer science from the University of Southwestern, Louisiana, in 1988. He is currently a professor in the Department of Computer Science and Information Engineering, National Cheng Kung University. His research interests include cryptology, network security, and coding theory.