Vol. 48 No. 9 IPSJ Journal Sep. 2007

Regular Paper

Ancestor Excludable Hierarchical ID-based Encryption and Its Application to Broadcast Encryption

Atsuko Miyaji†

An ID-based encryption (IBE) is a public key cryptosystem, in which a user's public key is given as a user ID. In IBE, only a single center generates all user secret keys, which may give the center a load of burdensome work. A hierarchical ID-based encryption (HIBE) is a kind of IBE and overcomes the problem by delegating a user secret key generation to a lower-level center, in which centers form a hierarchical structure. However, all ancestor nodes in HIBE act as centers. That is, any ancestor as well as the root can generate a secret key for any descendant node and, thus, a cipher text to a node can be decrypted by any ancestor node even if the ancestor does not have the same secret key as that of a target node. In this paper, we propose the concept of ancestor-excludable HIBE, in which ancestors with a level less than the designated one can be excluded from a set of privileged ancestors with a right to decrypt a cipher text to a target node. We also give the functional definition together with the security definition. This notion is denoted by AE-HIBE simply. We present the concrete example of AE-HIBE, which can work with constant-size ciphertext and decryption time, independent of the hierarchy level. We prove that our AE-HIBE is selective-ID-CPA secure in the standard model, which can be converted to be selective-ID-CCA secure by applying a general conversion method. Furthermore, AE-HIBE can be naturally applied to the broadcast encryption to realize the efficient public-key version with the user-key size of $O(\log^2 N)$ and the transmission rate of $O(r)$ for N users and r revoked users. The user-key size is the smallest at the transmission rate of $O(r)$, up to the present.

1. Introduction

IBE and HIBE. An ID-based encryption (IBE) is a public key cryptosystem, in which a user's public key is given as a user ID such as an email address or name and the corresponding secret key is computed by a center. In IBE, we don't have to use any certification for public keys. A concrete and provable secure IBE is proposed [3]. In IBE, only a single center generates all user secret keys, which may give the center a load of burdensome work. A hierarchical ID-based encryption (HIBE) is a kind of IBE and overcomes the problem by delegating a user secret key generation to a lower-level center: any user is assigned to a node of a tree except the root; the identity of a user v is an ID-tuple $v = v_1 \cdots v_{l-1} v_l$, which represents all ancestors information; and the corresponding user secret key $\mathtt{SK}_v$ is generated from its parent $v_1 \cdots v_{l-1}$. Therefore, we could say that HIBE represents ID hierarchically. However, all ancestor nodes act as centers. That is, any ancestor as well as root can generate a secret key for any descendant node and, thus, a cipher text to a node can be decrypted by any ancestor node even if the ancestor does not have the same secret key as that of a target node.

The feature of ID. To make the feature of ID clear, let us think about what information can identify a person from among others and what structure a user's identification data has. Let us think about the example of postal mail. In order to identify a person as Alice, we put a country name, a prefecture or state name, a city name, a street name, and a house number as well as her name. In the case of an e-mail, the information of an e-mail often consists of a country name, an organization name, and a department name as well as a user name such as Alice@computer-science.A-univ.edu. These examples indicate that ID itself forms a hierarchical structure. The hierarchical structure exists quite naturally in a variety of organizations, such as companies, governments and/or schools. We also see a rather relaxed hierarchical structure in a consortium of corporations that engage in governmental projects or joint ventures. In such a consortium, one company would be a leader and other companies would work in a part of the business as its sections, branches, or subsidiaries. Thus, such a consor-

† Japan Advanced Institute Science and Technology

This study is partly supported by Grant-in-Aid Scientific Research on Priority Area "Informatics" (Area #004) (16016242) and SCAT.

tium forms a hierarchical structure, but at the same time, each of the consortium companies is rather independent. In such a rather relaxed hierarchical structure, all upper-level members do not necessarily have to control the lower-level members. On the other hand, in a strict hierarchical structure, the upper-level member may absolutely control the lower-level member and even disclose ciphertexts sent to the lower-level member in the case of necessity.

Ancestor-excludable HIBE. Let us examine what features are necessary for IDs with a hierarchical structure. One feature is the *expression-ability of the hierarchical structure* of the ID, which can be achieved in HIBE but cannot be achieved in IBE. Another feature is the *ancestor-excludable feature*: ancestors with a level less than the designated one can be excluded from a set of privileged ancestors who have the right to decrypt a cipher text to a target node. All ancestors in HIBE act as centers and, thus, the upper-level member absolutely controls the lower-level member. Therefore, HIBE can not achieve the ancestor-excludable feature.

In summary, the ancestor-excludable hierarchical ID-based encryption scheme should satisfy the following two features: the expression-ability of the hierarchical structure of ID and the ancestor-excludable feature. IBE deals with ID at the same level, that is, it can neither express the hierarchical structure of ID nor realize the ancestor-excludable feature. On the other hand, HIBE can express the hierarchical structure of ID but cannot achieve the ancestor-excludable feature. We may note that the ancestor-excludable feature works well under the strict hierarchical structure as well as the relaxed hierarchical structure.

Our contribution. We propose the new concept of ancestor-excludable hierarchical ID-based encryption scheme, denoted by AE-HIBE for simplicity, and give the functional definition together with the security definition. We also give a concrete example of AE-HIBE, which is based on a HIBE [1]. Our AE-HIBE of the hierarchy level t can work with the constant-size ciphertext and decryption cost that is independent of t, and the user-key size of $O(t)$. Our AE-HIBE has proven to be selective-ID-CPA-secure in the standard model by introducing an injection map.

As an important application, our AE-HIBE improves the public-key subset difference broadcast encryption [6] in the user-key size at no expense of the transmission rate, which cannot be achieved in the HIBE paradigm. The subset difference broadcast encryption (SD) has an advantage of the transmission rate at the expense of user-key size, which is originally based on a symmetric key [10]. LSD broadcast encryption (LSD) [8] has further reduced the user-key size of SD at the slight expense of the transmission rate. HIBE translates the symmetric-key SD (resp. LSD) to the public-key one faithfully [6] and, thus, the user-key size is $O(|\mathtt{SK}_{\mathtt{HIBE}}| \log^2 N)$ (resp. $O(|\mathtt{SK}_{\mathtt{HIBE}}| \log^{3/2} N)$), where N is the number of users; and $|\mathtt{SK}_{\mathtt{HIBE}}|$ is the node-secret-key size in HIBE. Therefore, even if the most efficient HIBE (random-oracle model) [1] with $|\mathtt{SK}_{\mathtt{HIBE}}| = O(\log N)$ for the hierarchy level $\log N$ is applied, the user-key size of public-key SD (resp. LSD)* becomes $O(\log^3 N)$ (resp. $O(\log^{5/2} N)$). On the other hand, the ancestor-excludable feature of AE-HIBE can exactly reduce the user-key size by realizing another feature of SD. In fact, the user-key size in the public-key SD based on AE-HIBE is $O(|\mathtt{SK}_{\mathtt{AE\text{-}HIBE}}| \log N)$, where $|\mathtt{SK}_{\mathtt{AE\text{-}HIBE}}|$ is the node-secret-key size in AE-HIBE. As a result, the public-key SD based on our AE-HIBE** can work with the user-key size of $O(\log^2 N)$, which is the smallest at no expense of the transmission rate, up to the present. See **Table 1** for the comparison.

This paper is organized as follows. Section 2 summarizes the basic notions. Section 3 gives the functional definition of AE-HIBE and the security definition. Section 4 presents a concrete example together with the security proof. Finally, Section 5 applies our AE-HIBE to the broadcast encryption and presents an efficient public-key broadcast encryption.

2. Preliminary

This section summarizes the basic notions and HIBE.

2.1 The Bilinear Map and Its Related Assumption

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime

* The number of keys that a user has to possess is $O(\log^2 N)$ (resp. $O(\log^{3/2} N)$), where the size of a key is $O(\log N)$. So the total user-key size becomes $O(\log^3 N)$ (resp. $O(\log^{5/2} N)$).

** The number of keys that a user has to possess is $O(\log N)$, where the size of a key is $O(\log N)$. So the total user-key size becomes $O(\log^2 N)$.

Table 1 Comparison of public-key broadcast encryption with N users and r revoked users.

	CS [6]+IBE [3]	SD [6]+HIBE [1]	LSD [6]+HIBE [1]	BGW-method[☆2)]	Ours
transmission rate	$O(r \log \frac{N}{r})$	$O(r)$	$O(r)$	$O(1)^{\dagger}$	$O(r)$
user-key size	$O(\log N)$	$O(\log^3 N)$	$O(\log^{5/2} N)$	$O(1)$	$O(\log^2 N)$
public-key size	$O(1)$	$O(1)^{\ddagger}$	$O(1)^{\ddagger}$	$O(N)^{\dagger}$	$O(1)^{\ddagger}$
decryption time	$O(1)$	$O(\log N)$	$O(\log N)$	$O(N-r)$	$O(\log N)$

†: The public-key size can be reduced, while maintaining |public key| × |transmission rate| = $O(N)$.
‡: The public-key sizes can be made constant size under the random oracle model.

order q. $\mathbb{G}_1$ (resp. $\mathbb{G}_2$) is represented additively (resp. multiplicatively), where $\mathcal{O}$ (resp. 1) represents the zero element (identity element) for addition (multiplication) in $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). The following bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is defined over $\mathbb{G}_1$.

(1) Bilinearity: $\hat{e}(xP_0, yP_1) = \hat{e}(yP_0, xP_1) = \hat{e}(P_0, P_1)^{xy}$ for all $P_0, P_1 \in \mathbb{G}_1$ and all $x, y \in \mathbb{Z}_q$.
(2) Non-degeneracy: $\hat{e}(P, P) \neq 1$ for any $P \in \mathbb{G}_1 \setminus \{\mathcal{O}\}$.
(3) Computability: There is an efficient algorithm to compute $\hat{e}(P_0, P_1)$ for any $P_0, P_1 \in \mathbb{G}_1$.

Let k be a security parameter. A BDHE (Bilinear Diffie-Hellman Exponent) parameter generator $\mathcal{IG}$ is a probabilistic polynomial time (PPT) algorithm that on input 1^k, outputs a description of the above $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$. The *computational ℓ-BDHE problem* with respect to $\mathcal{IG}$ is to compute $\hat{e}(P, R)^{\alpha^\ell}$ from random P, R and $\alpha^i P \in \mathbb{G}_1$ with $i = 1, \cdots, \ell-1, \ell+1, \cdots, 2\ell$, where $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$ is an output of $\mathcal{IG}$. If this problem is hard, then we say that $\mathcal{IG}$ satisfies the *computational ℓ-BDHE assumption*. More precisely, we say that $\mathcal{IG}$ satisfies the *computational ℓ-BDHE assumption* if the following probability is negligible (in k) for all PPT algorithms A:

$$\Pr\left[\begin{array}{l}(\mathbb{G}_1, \mathbb{G}_2, \hat{e}) \leftarrow \mathcal{IG}(1^k); P, R \leftarrow \mathbb{G}_1; \\ \alpha \leftarrow \mathbb{Z}_q : \\ A(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, R, Y_1, \cdots, Y_{\ell-1}, \\ \ Y_{\ell+1}, \cdots, Y_{2\ell}) = \hat{e}(P, R)^{\alpha^\ell}\end{array}\right],$$

where $Y_i = \alpha^i P$. The *decisional* version is defined in the usual manner: given random $P, R, \alpha^i P \in \mathbb{G}_1$ and $T \in \mathbb{G}_2$ (where $i = 1, \cdots, \ell-1, \ell+1, \cdots, 2\ell$), decide whether $T = \hat{e}(P, R)^{\alpha^\ell}$ or not. More precisely, we say that $\mathcal{IG}$ satisfies the *decisional ℓ-BDHE assumption* if the following is negligible (in k) for all PPT algorithms A:

$$\left|\Pr\left[\begin{array}{l}(\mathbb{G}_1, \mathbb{G}_2, \hat{e}) \leftarrow \mathcal{IG}(1^k); \\ P, R \leftarrow \mathbb{G}_1; \alpha \leftarrow \mathbb{Z}_q : \\ A(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, R, Y_1, \cdots, Y_{\ell-1}, \\ \ Y_{\ell+1}, \cdots, Y_{2\ell}, \\ \ \hat{e}(P, R)^{\alpha^\ell}) = 0\end{array}\right]\right.$$
$$\left.-\Pr\left[\begin{array}{l}(\mathbb{G}_1, \mathbb{G}_2, \hat{e}) \leftarrow \mathcal{IG}(1^k); \\ P, R \leftarrow \mathbb{G}_1; \alpha \leftarrow \mathbb{Z}_q; \\ T \leftarrow \mathbb{G}_2 : \\ A(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, R, Y_1, \cdots, Y_{\ell-1}, \\ \ Y_{\ell+1}, \cdots, Y_{2\ell}, T) = 0\end{array}\right]\right|$$

These assumption are believed to hold if $\hat{e}$ is Weil/Tate pairing on supersingular elliptic curves or some ordinary elliptic curves [9].

2.2 ID-based Encryption and Hierarchical ID-based Encryption

In the model of HIBE, we consider a tree of hierarchy, where the root node is labeled with $v_0 = \epsilon$ (See Fig. 2). The *identity* of a user v is represented as an ID-tuple $v = v_1 \cdots v_h \cdots v_l$, where $v_1, \cdots, v_l$ are the nodes on the path from v_0 to v_l of the tree. A child node (or an ancestor node) of v is represented as $vv_{l+1} = v_1 \cdots v_l v_{l+1}$ (or $v|_h = v_1 \cdots v_h$) by borrowing notation [5]. A secret-key of the identity v, denoted by $\mathtt{SK}_v$, is issued by its parent (that is $v|_{l-1}$). IBE can be considered for the case in which all identities are in level 1 of HIBE (See Fig. 1). The formal definition of IBE or HIBE is summarized in Annex A.1 or A.2, respectively.

3. Ancestor-Excludable Hierarchical ID-based Encryption

This section defines the ancestor-excludable hierarchical ID-based encryption (AE-HIBE). AE-HIBE is the intermediate notion between IBE and HIBE, which achieves a new feature, the *ancestor-excludable* feature.

3.1 Comparisons between IBE, HIBE, and AE-HIBE

Let us discuss the differences between IBE, HIBE, and AE-HIBE to clarify an issue to be settled. **Figures 1** and **2** show the relation between centers and users in IBE and HIBE, re-

☆ This method is not based on SD or LSD.

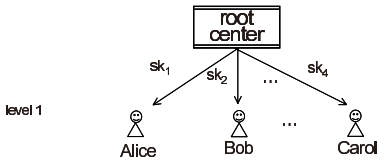

Fig. 1 IBE.

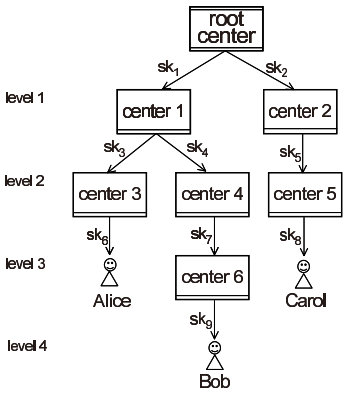

Fig. 2 HIBE.

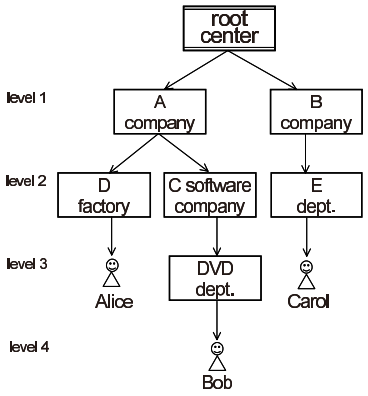

Fig. 3 AE-HIBE.

spectively. In IBE, there exists only one center that generates all secret keys for all users. So IBE deals with users at the same level. On the other hand, centers form a hierarchical structure in HIBE. Therefore, HIBE represents the ID hierarchically. This feature is called the *expression-ability of the hierarchical structure* of IDs in this paper.

The hierarchical structure exists quite naturally in various organizations such as companies, governments and/or schools, where the ID itself often forms a hierarchical structure. Our purpose is to construct the ID-based encryption scheme fit for IDs with a hierarchical structure. Let us consider the following situation in **Fig. 3**: Alice works in the D factory of A company; Bob works in the DVD department of C software company under A company. C software company has been recently absorbed by A company. To identify Alice among others, we need all hierarchical information of (A company, D factory, Alice) as well as her name, Alice. As we mentioned above, IBE is not suitable for this case. Let us apply HIBE (see Fig. 2) to this situation: the root center generates the secret keys for A company and B company, the president of A company generates the secret keys for D factory and C software company, and the manager of D factory generates the secret key for Alice, and so on. HIBE seems to deal with the hierarchical structure and represents the hierarchical structure of the ID. However, all nodes except leaves in HIBE act as centers, so a cipher text to Alice can be decrypted by both the manager of D factory and the president of A company as well as the root center. In a strict hierarchical structure, the upper-level member may be allowed to disclose ciphertexts to the lower-level member if necessity, however, in a rather relaxed hierarchical structure, no upper-level member necessarily has to control the lower level. In the case of Bob, C software company has been recently absorbed by A company. So, C software company is in the position under A company, but at the same time, is rather independent of A company. In this case, only the designated upper-level member such as the managers of C software company or DVD Department is able to control the lower-level member. This feature is called the *ancestor-excludable feature* in this paper. The ancestor-excludable feature can also achieve the original feature of HIBE, that is the upper-level member always controls the lower-level member, by setting the level of a designated ancestor to 1.

In summary, AE-HIBE satisfies both the expression-ability of the hierarchical structure of ID and the ancestor-excludable feature. Neither HIBE nor IBE can achieve the ancestor-excludable feature.

3.2 Functional and Security Definition

This section gives the functional definition of AE-HIBE and its security definition. Let $\mathcal{T}$ be an m-ary tree with the level t, which represents user IDs. Let us denote a restricted set by $S|_h = \{s_1, \cdots, s_h\}$ for a set $S = \{s_1, \cdots, s_h, \cdots, s_l\}$

Definition 1 (AE-HIBE) AE-HIBE consists of a 5-tuple of PPT algorithms (`KGen`, `KDer`, $\texttt{KDer}_p$, `Enc`, `Dec`), where

- $\texttt{KGen}(1^k, t, m)$, the root center key-genera-

tion algorithm, for input of security parameter k and the level t of m-ary tree, outputs a system public key PK and the root center's secret key $\mathtt{SK}_\varepsilon$.

- $\mathtt{KDer}(\mathsf{PK}, v, \mathtt{SK}_\varepsilon)$, the root center key-derivation algorithm, for input of the public key PK, a node v whose level is l, and the root secret key $\mathtt{SK}_\varepsilon$, outputs the node secret key of v, $\mathtt{SK}_v = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,l}\}$.
- $\mathtt{KDer}_p(\mathsf{PK}, v, \mathtt{SK}_v, vv_{l+1})$, the user's key derivation algorithm, for input of the public key PK, a node v with level l, the secret key $\mathtt{SK}_v = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,l}\}$, and a child vv_{l+1} outputs part of node secret keys of the child vv_{l+1}. Note that v can derive only $\mathtt{SK}_{vv_{l+1}}|_l = \mathtt{SK}_{vv_{l+1}} \setminus \{\mathtt{sk}_{vv_{l+1},l+1}\}$ but cannot derive $\mathtt{sk}_{vv_{l+1},l+1}$. Only the root center can derive all node secret keys of users.
- $\mathtt{Enc}(\mathsf{PK}, v, h, M)$, the encryption algorithm, for input of the public key PK, a user ID $v = v_1 \cdots v_h \cdots v_l$, level h of a designated ancestor, and a message M, outputs a ciphertext C together with the user ID v and the level h.
- $\mathtt{Dec}(C, v, h, \mathtt{SK}_v|_i) = M (h \leq i \leq l)$, the decryption algorithm, for input of a ciphertext C, a user ID $v = v_1 \cdots v_h \cdots v_i \cdots v_l$, level h of a designated ancestor, and a subset of a user secret key $\mathtt{SK}_v|_i$, decrypts C to M.

Remark:
1. The advantage of AE-HIBE to HIBE lies in a feature of Dec. In the case of HIBE, all ancestors of v can decrypt a ciphertext. However, AE-HIBE can exclude any ancestor with a level $i < h$ and let only ancestors with levels $i \geq h$ decrypt the ciphertext. Note that an ancestor with a level of $i \geq h$ cannot generate the whole key of $\mathtt{SK}_v$ but $\mathtt{SK}_v|_i$ by executing $\mathtt{KDer}_p$.
2. Because of the above feature, the key structure of $v = v_1 \cdots v_l$ becomes $\mathtt{SK}_v = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,l}\}$. Any ciphertext to v with the designated level h needs a subset $\mathtt{SK}_v|_h = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,h}\}$, which cannot be structurally generated by any ancestor with the level $i < h$. The correctness of AE-HIBE is defined as follows.

Definition 2 (Correctness) Let AE-HIBE be the ancestor-excludable HIBE in Definition 1. AE-HIBE satisfies the correctness if the following features hold. Let C be a cipher text to a target node v with the designated level h, that is $C = \mathtt{Enc}(\mathsf{PK}, v, h, M)$. Then
1. Any ancestor of v with a level $\geq h$ can decrypt the cipher text.
2. Any ancestor of v with a level $< h$ cannot decrypt the cipher text.

AE-HIBE is a special case of HIBE and, thus, the security definition follows mostly that of HIBE [1)], which has the decryption and the key derivation oracles. The important difference lies in the key derivation oracles. In the case of HIBE, an adversary is not allowed to ask a secret key of any node in the path $\rho_v \in \mathcal{T}$ from the root to a target node v. But, in our AE-HIBE, an adversary is allowed to ask a secret key of a node w in the path ρ_v until the level of w is lower than the target level. The security is defined as follows.

Definition 3 We say that an AE-HIBE scheme is IND-AE-HIBE-CCA secure against adaptive chosen ciphertext and node adversary if the advantage of any PPT adversary A against the challenger in the following experiment is negligible.

Set up The challenger takes a security parameter k and the level t of an m-ary tree and executes $\mathtt{KGen}(1^k, t, m)$. Then it gives A the public parameter PK and keeps the root secret key $\mathtt{SK}_\varepsilon$.

Phase 1 A issues a number of queries $q_1, \cdots, q_{n_1}$, where query q_i is one of the following:

- Node-secret-key query: On the query of a node v, output the corresponding node key $\mathtt{SK}_v$.
- Decryption query: On the query of a node $v = v_1 \cdots v_h \cdots v_l$, the target level h, and a ciphertext C, output the recovered message M.

Challenge A outputs two equal length messages $M_0, M_1 \in \{0,1\}^*$, a node v^*, and a target level h^*. The only constraint is that A did not previously issue a node-secret-key query on $v^*|_i$ with $i \geq h^*$ for the target node v^*. Then the challenger picks a random bit $b \in \{0,1\}$, sets $C^* = \mathtt{Enc}(\mathsf{PK}, v^*, h^*, M_b)$, and sends C^* to A as a challenge.

Phase 2 A continues a number of queries $q_{n_1+1}, \cdots, q_n$, where a query q_i is one of the following in the same way as Phase 1:

- Node-secret-key query: On the query of v under the constraint in Challenge, output the corresponding node key $\mathtt{SK}_v$.
- Decryption query: On the query of $(v, h, C) \notin \{(v^*, i, C^*) | i \geq h^*\}$, output

the recovered message M.

Guess A outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b = b'$. The advantage of A attacking the scheme is defined as $|\Pr[b = b'] - 1/2|$.

A weaker security of IND-AE-HIBE-CCA called a selective-node chosen ciphertext secure AE-HIBE (IND-sAE-HIBE-CCA) is defined in the same way as HIBE [1]. The game is the same as IND-AE-HIBE-CCA except that the adversary A discloses to the challenger the target node v^* and the level h^* before the Set up phase. The Node-secret-key query follows the restrictions in Phase 2. We can also define the chosen plaintext security for an AE-HIBE scheme in the same way as IBE or HIBE [1),3),7)], in which A is not allowed to issue any decryption query but still issues adaptive node-secret-key queries. This adversary function is termed AE-HIBE-CPA (or sAE-HIBE-CPA in the case of a selective-node adversary).

Definition 4 We say that an AE-HIBE scheme is IND-sAE-HIBE-CCA secure against adaptive chosen ciphertext and a selective-node adversary if the advantage of any PPT adversary A against the challenger in the experiment defined in Definition 3 is negligible.

Definition 5 We say that an AE-HIBE scheme is IND-AE-HIBE-CPA (resp. IND-sAE-HIBE-CPA) secure against adaptive chosen node (resp. selective-node) attacks if the advantage of any PPT adversary A against the challenger in the experiment defined in Definition 3 without decryption oracle is negligible.

3.3 Discussion

Let us investigate why we need the ancestor-excludable feature for HIBE. One of important applications of HIBE is the broadcast encryption scheme. The efficiency of broadcast encryption is measured by the transmission rate and the size of a user secret key, however, here we focus on only the size of a user secret key.

HIBE is used to realize the public-key subset-difference broadcast encryption scheme. The subset difference broadcast encryption (SD) is originally based on a symmetric-key encryption [10)], in which users are assigned to leaves of a binary tree $\mathcal{T}$ of level t for $|\mathcal{N}| = N = 2^t$. In SD, a set of privileged users is covered by the difference of two subsets $S_{v,w} = S_v \setminus S_w$, where $v, w \in \mathcal{T}$, v is an ancestor of w, and S_v or S_w is a complete subtree rooted at v or w, respectively. **Figure 4** represents a set of privileged users of SD, in which privileged users are in the gray-color part and revoked users are in the colorless part. The remarkable feature of SD lies in the user key derivation methods, which enables a user key of $O(\log^2 N)$ to generate the whole necessary decryption keys with $O(N)$. HIBE translates the symmetric-key SD to the public-key faithfully and, thus, the user-key size is $O(|\mathtt{SK}_{\mathtt{HIBE}}| \log^2 N)$. Therefore, even if the most efficient HIBE [1)] or the limited-delegation version (see Section 4.1 in the same paper [1)]) is applied, the user-key size becomes $O(\log^3 N)$, which is worse than the user-key size $O(\log^2 N)$ in the symmetric-key SD.

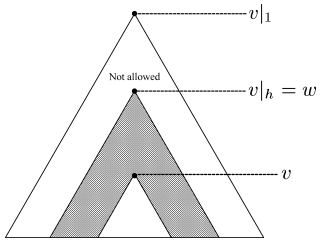

Fig. 4 A set of privileged users of SD.

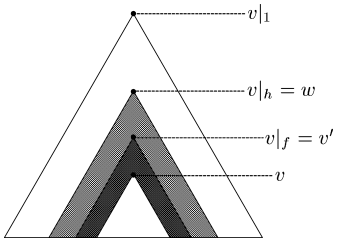

Fig. 5 Ancestor-excludable feature in SD ($S_{w,v} \supsetneqq S_{v',v}$).

In order to improve the user-key size, we investigate a new key-derivation method of SD. The covering method of SD satisfies the following feature, which is focused in this paper for the first time: $S_{w,v} \supsetneqq S_{v',v}$($v = v_1 \cdots v_h \cdots v_f \cdots v_l$, $w = v|_h$, and $v' = v|_f$). See **Fig. 5**. The feature means that a user in $S_{v',v}$ is allowed to generate a subset key $K_{S_{w,v}}$ but a user in $S_{w,v} \setminus S_{v',v}$ is not allowed to generate a subset key $K_{S_{v',v}}$. The ancestor-excludable feature would achieve this feature by excluding any ancestor with a level $> v'$ from users in $S_{w,v}$. The above discussion will be described in detail in Section 5.

4. ID-based Encryption Scheme with a Hierarchical Structure

We present a concrete scheme of AE-HIBE and, then, give security proof. The secu-

rity of our AE-HIBE is based on the decisional t-BDHE assumption without random oracle model, where t is the level of an m-ary tree $\mathcal{T}$ with the root ε. The identity of a user v is represented as an ID-tuple $v = v_1 \cdots v_h \cdots v_l$, where $v_1, \cdots, v_l$ are the nodes on the path from v_0 to v_l of $\mathcal{T}$.

4.1 AE-HIBE Based on the BDHE Assumption

Our AE-HIBE makes use of HIBE proposed by Boneh, Boyen, and Goh [1], called BBG-HIBE in this paper. In order to give the security proof, we newly introduce an injection map from a user $v = v_1 \cdots v_l \in \mathcal{T}$ to $\mathbb{Z}_q^*$,

$$i_d : \mathcal{T} \to \mathbb{Z}_q^*.$$

The number of nodes is $\frac{m^{t+1}-1}{m-1} \leq m^{t+1} - 1$. The injection map can be defined whenever $m^{t+1} - m < q - 1$: for example, i_d is defined by numbering v from $i_d(\varepsilon) = 1$ according to some pre-defined order☆. Then, $v \neq v'$ if and only if $i_d(v) \neq i_d(v')$ for $v = v_1 \cdots v_j, v' = v'_1 \cdots v'_j \in \mathcal{T}$ and $i_d(v) \neq 0, 1$ for $v = v_1 \cdots v_h \in \mathcal{T}$ correspond to a user v are satisfied.

In our AE-HIBE, a message to be encrypted is in $\mathbb{G}_2$ in the same way as[1] but identities $v = v_1 \cdots v_l$ are in $\{0,1\}^*$ unlike [1]☆☆ by using the above injection map i_d.

A secret key of a user v will consist of $(t+1)$ group elements, denoted by $\mathtt{SK}_v = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,l}\}$, where $\mathtt{sk}_{v,1} = \{A_{v,1}, B_v, C_{v,l+1}, \cdots, C_{v,t}\}$ and $\mathtt{sk}_{v,i} = A_{v,i} (i = 2, \cdots, l)$.

$\mathtt{KGen}(1^k, t)$ executes the following:

(1) Run $\mathcal{IG}(k)$ to generate groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with prime order q and bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
(2) Set an injection map, $i_d : \mathcal{T} \to \mathbb{Z}_q^*$.
(3) Choose random points $P, P_1, Q_1, \cdots, Q_t, R_1, \cdots, R_t \in \mathbb{G}_1$ and a random secret $\alpha \in \mathbb{Z}_q$.
(4) Compute $Q = \alpha P$ and the root secret key $\mathtt{SK}_{\varepsilon,i} = \alpha R_i$ for $1 \leq i \leq t$.
(5) The public key is

$$\mathsf{PK} = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, i_d, P, P_1, Q, Q_1, \cdots, Q_t, R_1, \cdots, R_t\}$$

and the root secret key is

$$\mathtt{SK}_\varepsilon = \{\mathtt{SK}_{\varepsilon,1}, \cdots, \mathtt{SK}_{\varepsilon,t}\}.$$

$\mathtt{KDer}(\mathsf{PK}, v, \mathtt{SK}_\varepsilon)$ executes the following:

(1) Let $v = v_1 \cdots v_l$.
(2) Choose a random secret $\alpha_v \in \mathbb{Z}_q$ and compute

$$\begin{aligned} A_{v,i} &= \mathtt{SK}_{\varepsilon,i} + \alpha_v(i_d(v|_i)Q_i \\ &\quad + i_d(v|_{i+1})Q_{i+1} + \cdots \\ &\quad + i_d(v)Q_l + P_1) \ (i = 1, \cdots, l), \\ B_v &= \alpha_v P, \\ C_{v,j} &= \alpha_v Q_j \ (l+1 \leq j \leq t) \end{aligned}$$

(3) Output $\mathtt{SK}_v = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,l}\}$, where $\mathtt{sk}_{v,1} = \{A_{v,1}, B_v, C_{v,l+1}, \cdots, C_{v,t}\}$, $\mathtt{sk}_{v,i} = A_{v,i} \ (i = 2, \cdots, l)$

$\mathtt{KDer}_p(\mathsf{PK}, v, \mathtt{SK}_v, vv_{l+1})$ executes the following:

(1) Let $v = v_1 \cdots v_l$.
(2) Parse $\mathtt{SK}_v = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,l}\}$, where

$$\begin{aligned} \mathtt{sk}_{v,1} &= \{A_{v,1}, B_v, C_{v,l+1}, \cdots, C_{v,t}\}, \\ \mathtt{sk}_{v,i} &= A_{v,i} \ (i = 2, \cdots, l) \\ A_{v,i} &= \mathtt{SK}_{\varepsilon,i} + \alpha_v\Big(i_d(v|_i)Q_i \\ &\quad + i_d(v|_{i+1})Q_{i+1} + \cdots + i_d(v)Q_l \\ &\quad + P_1\Big) \ (i = 1, \cdots, l), \\ B_v &= \alpha_v P, \text{ and} \\ C_{v,j} &= \alpha_v Q_j \ (l+1 \leq j \leq t). \end{aligned}$$

(3) Choose a random secret $r_{vv_{l+1}} \in \mathbb{Z}_q$.
(4) Compute

$$\begin{aligned} A_{vv_{l+1},i} &= A_{v,i} + i_d(vv_{l+1})C_{v,l+1} \\ &\quad + r_{vv_{l+1}}(i_d(v|_i)Q_i \\ &\quad + i_d(v|_{i+1})Q_{i+1} + \cdots \\ &\quad + i_d(v)Q_l + i_d(vv_{l+1})Q_{l+1} \\ &\quad + P_1) \ (i = 1, \cdots, l), \\ B_{vv_{l+1}} &= B_v + r_{vv_{l+1}}P, \text{ and} \\ C_{vv_{l+1},j} &= C_{v,j} + r_{vv_{l+1}}Q_j \\ &\quad (j = l+2, \cdots, t). \end{aligned}$$

(5) Set

$$\mathtt{sk}_{vv_{l+1},1} = \{A_{vv_{l+1},1}, B_{vv_{l+1}}, C_{vv_{l+1},l+2}, \cdots, C_{vv_{l+1},t}\}$$

and $\mathtt{sk}_{vv_{l+1},i} = \{A_{vv_{l+1},i}\} \ (2 \leq i \leq l)$. Then, it becomes a valid part of the key of vv_{l+1} for $\alpha_{vv_{l+1}} = \alpha_v + r_{vv_{l+1}}$, which is unknown to user v.
(6) Output $\{\mathtt{sk}_{vv_{l+1},1}, \cdots, \mathtt{sk}_{vv_{l+1},l}\}$.

$\mathtt{Enc}(\mathsf{PK}, v, h, M)$ executes the following:

(1) Let $v = v_1 \cdots v_h \cdots v_l$.
(2) Choose a random $\gamma \in \mathbb{Z}_q$.
(3) Compute $C = \{M \cdot d, \gamma P, \gamma(i_d(v|_h)Q_h + i_d(v|_{h+1})Q_{h+1} + \cdots + i_d(v)Q_l + P_1)\}$, where $d = \hat{e}(Q, R_h)^\gamma$.
(4) Output $\{C, v, h\}$.

$\mathtt{Dec}(\mathtt{SK}_v, C, h)$ executes the following:

☆ One example is a pre-order traversal.
☆☆ identities in Ref. 1) have to be in $\mathbb{Z}_{q^l}$.

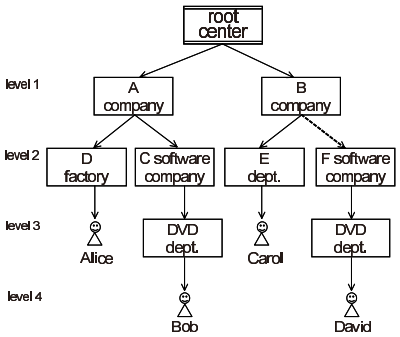

Fig. 6 Addition of a new organization.

(1) Let $v = v_1 \cdots v_h \cdots v_l$ and $C = \{C_1, C_2, C_3\}$.
(2) Parse $\mathtt{sk}_{v,1} = \{A_{v,1}, B_v, C_{v,l+1}, \cdots, C_{v,t}\}$, $\mathtt{sk}_{v,i} = A_{v,i}$ $(i = 2, \cdots, l)$.
(3) Compute $M = C_1/d$, where $d = \hat{e}(C_2, A_{v,h})/\hat{e}(B_v, C_3)$.

The decryption succeeds as follows,

$$\frac{\hat{e}(C_2, A_{v,h})}{\hat{e}(B_v, C_3)} = \frac{\hat{e}(\gamma P, \mathtt{SK}_{\varepsilon,h} + \alpha_{v,h}(i_d(v|_h)Q_h + \cdots + i_d(v)Q_l + P_1)))}{\hat{e}(\alpha_{v,h}P, \gamma(i_d(v|_h)Q_h + \cdots + i_d(v)Q_l + P_1))} = \hat{e}(Q, R_h)^{\gamma}.$$

Both $A_{v,h}$ and B_v in $\mathtt{SK}_v$ is sufficient to decrypt a ciphertext $C \leftarrow \mathtt{Enc}(\mathsf{PK}, v, h, M)$. Therefore, only ancestors of v with a level $\geq h$ can decrypt the C since no ancestor of v with a level $< h$ has any information about $\mathtt{sk}_{v,h}$. Thus, AE-HIBE has proven to satisfy the ancestor-excludable feature as well as the expression-ability of the hierarchical structure of ID.

Remark: In addition to the above two features, our AE-HIBE has another interesting feature on an addition of a new organization. Suppose that B company absorbs F software company with DVD Dept. (see **Fig. 6**) after the root center has already set up each secret key (see Fig. 3), where David works in the DVD Dept. In that case, the root center can easily add a new group of (F software company, DVD Dept., David) by setting them in $\mathcal{T}$ and executing `KDer` to generate each secret key. Then, the ancestor-excludable feature still holds in the new hierarchical structure if the addition satisfies the following conditions: the new structure does not break the original structure and the number of total nodes does not exceed $q - 1$. Note that the root center does not have to regenerate the secret keys of any user in Fig. 3.

4.2 Security

We show that our AE-HIBE is selective-identity secure (`IND-AE-HIBE-CPA`) under the decisional Bilinear Diffie-Hellman Exponent assumption.

Theorem 1 If $\mathcal{IG}$ satisfies the $(t+1)$-BDHE assumption, then AE-HIBE is `IND-sAE-HIBE-CPA` secure.

proof: Suppose there exists a `ppt` adversary A which attacks the proposed AE-HIBE of an m-ary tree $\mathcal{T}$ with hierarchy level t with the advantage ε, where A asks Q_e secret-key queries. We will show that a `ppt`-algorithm B exists that solves the decisional $(t+1)$-BDHE problem in $\mathbb{G}_1$.

For $P \in \mathbb{G}_1$ and $\alpha \in \mathbb{Z}_q$, let $Y_i = \alpha^i P$. B is given an output $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$ of $\mathcal{IG}(1^k)$ and a random tuple $(P, R, Y_1, \cdots, Y_t, Y_{t+2}, \cdots, Y_{2t+2}, T)$ that is either sampled from $\mathcal{P}_{\mathtt{BDHE}}$ (where $T = e(P,R)^{\alpha^{t+1}}$) or from $\mathcal{R}_{\mathtt{BDHE}}$ (where T is uniform and independent in $\mathbb{G}_2$). The goal of B is to determine whether $T = e(P,R)^{\alpha^{t+1}}$ or not. B executes A in a selective identity game as follows.

Initialization: A first outputs an identity $v^* = v_1^* \cdots v_l^* \in \{0,1\}^*$ of level l $(l \leq t)$ and a designated level h^* $(h^* \leq l)$ that A tries to attack. If $l \leq t$, then B appends random elements to v^* until the level of v^* is t and keeps these extra values to itself. Hereafter we assume that the level of v^* is t.

Setup: To generate the public key PK, B executes the following.

- Set $Q = Y_1 = \alpha P$, choose $\gamma_1, \cdots, \gamma_t \in \mathbb{Z}_q$ randomly and set
$$Q_i = \gamma_i P - Y_{t-i+1} = (\gamma_i - \alpha^{t-i+1})P$$
for $1 \leq i \leq t$.
- Choose $\delta \in \mathbb{Z}_q$ randomly and set $P_1 = \delta P + \sum_{i=h^*}^{t} i_d(v^*|_i) Y_{t-i+1}$.
- Choose $\beta_1, \cdots, \beta_t \in \mathbb{Z}_q$ randomly and set
$$R_i = Y_t + \beta_i P = (\beta_i + \alpha^t)P,$$
$$\mathtt{SK}_{\varepsilon,i} = \alpha R_i = Y_{t+1} + \beta_i Y_1 \ (1 \leq i \leq t),$$
where $\mathtt{SK}_{\varepsilon,i}$ is unknown to B.
- Output $\mathsf{PK} = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, i_d, P, P_1, Q, Q_1, \cdots, Q_t, R_1, \cdots, R_t\}$ and $\mathtt{SK} = \{\mathtt{SK}_{\varepsilon,1}, \cdots, \mathtt{SK}_{\varepsilon,t}\}$, and sends PK to A.

Phase 1: A issues a private key corresponding to $v = v_1 \cdots v_u \in \mathcal{T}$ where $u \leq t$. The only restriction is that $i_d(v) \neq i_d(v^*|_i)$ for $h^* \leq i \leq l$. To respond to the query, B derives a private key according to the following two cases.

∘ `Case 1` $u < h^*$: B executes the following. Note that $i_d(v) \neq 0$ from the definition of i_d.

- Choose $\tilde{\alpha}_v \in \mathbb{Z}_q$ randomly and set

$$\begin{aligned}\alpha_v &= \frac{\alpha^u}{i_d(v)} + \tilde{\alpha}_v \\ B_v &= \alpha_v P = \frac{\alpha^u}{i_d(v)} P + \tilde{\alpha}_v P \\ &= \frac{1}{i_d(v)} Y_u + \tilde{\alpha}_v P \\ C_{v,j} &= \alpha_v Q_j \\ &= \frac{\alpha^u}{i_d(v)}(\gamma_j P - Y_{t-j+1}) + \tilde{\alpha}_v Q_j \\ &= \frac{1}{i_d(v)}(\gamma_j Y_u - Y_{t-j+u+1}) \\ &\quad + \tilde{\alpha}_v Q_j \ (u+1 \leq j \leq t).\end{aligned}$$

  All these parameters except α_v are computable by B.
- For $A_{v,i}$ $(1 \leq i \leq u)$, set

$$\begin{aligned}A_{v,i} &= \mathtt{SK}_{\varepsilon,i} + \alpha_v(i_d(v|_i)Q_i + \cdots \\ &\quad + i_d(v)Q_u + P_1) \\ &= \mathtt{SK}_{\varepsilon,i} \\ &\quad + \frac{1}{i_d(v)} \sum_{j=i}^{u} i_d(v|_j)(\gamma_j Y_u - Y_{t+u-j+1}) \\ &\quad + \tilde{\alpha}_v \sum_{j=i}^{u} i_d(v|_j)(\gamma_j P - Y_{t-j+1}) \\ &\quad + \frac{\delta}{i_d(v)} Y_u + \tilde{\alpha}_v \delta P \\ &\quad + \sum_{j=h^*}^{t} i_d(v^*|_j)\left(\frac{Y_{t-j+u+1}}{i_d(v)} + \tilde{\alpha}_v Y_{t-j+1}\right),\end{aligned}$$

  all of which, except $\mathtt{SK}_{\varepsilon,i} - Y_{t+1}$, are computable by B. The exception is computable as follows:

$$\begin{aligned}&\mathtt{SK}_{\varepsilon,i} - Y_{t+1} \\ &= Y_{t+1} + \beta_i Y_1 - Y_{t+1} = \beta_i Y_1.\end{aligned}$$

  Therefore, B can compute $A_{v,i}$.
- Output $\mathtt{SK}_v = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,u}\}$ to A, where $\mathtt{sk}_{v,1} = \{A_{v,1}, B_v, C_{v,u+1}, \cdots, C_{v,t}\}$, $\mathtt{sk}_{v,i} = A_{v,i}$ $(i = 2, \cdots, u)$.

∘ `Case 2` $u \geq h^*$: The restriction of private key queries ensures that $i_d(v^*|_j) \neq i_d(v)$ for $h^* \leq j \leq l$. Then it holds that $i_d(v^*|_u) \neq i_d(v)$. B executes the following:

- Choose $\tilde{\alpha}_v \in \mathbb{Z}_q$ randomly and set

$$\begin{aligned}\alpha_v &= \frac{\alpha^u}{i_d(v) - i_d(v^*|_u)} + \tilde{\alpha}_v \\ B_v &= \alpha_v P = \frac{\alpha^u}{i_d(v) - i_d(v^*|_u)} P + \tilde{\alpha}_v P \\ &= \frac{1}{i_d(v) - i_d(v^*|_u)} Y_u + \tilde{\alpha}_v P \\ C_{v,j} &= \alpha_v Q_j \\ &= \frac{\alpha^u}{i_d(v) - i_d(v^*|_u)}(\gamma_j P - Y_{t-j+1}) \\ &\quad + \tilde{\alpha}_v Q_j \\ &= \frac{1}{i_d(v) - i_d(v^*|_u)}(\gamma_j Y_u - Y_{t-j+u+1}) \\ &\quad + \tilde{\alpha}_v Q_j \ (u+1 \leq j \leq t).\end{aligned}$$

  All these parameters except α_v are computable by B.
- For $A_{v,i}$ $(1 \leq i \leq u)$, set

$$\begin{aligned}A_{v,i} &= \mathtt{SK}_{\varepsilon,i} + \alpha_v(i_d(v|_i)Q_i + \cdots \\ &\quad + i_d(v)Q_u + P_1) \\ &= \mathtt{SK}_{\varepsilon,i} \\ &\quad + \frac{1}{i_d(v) - i_d(v^*|_u)} \sum_{j=i}^{u} i_d(v|_j)(\gamma_j Y_u - Y_{t+u-j+1}) \\ &\quad + \tilde{\alpha}_v \sum_{j=i}^{u} i_d(v|_j)(\gamma_j P - Y_{t-j+1}) \\ &\quad + \frac{\delta}{i_d(v) - i_d(v^*|_u)} Y_u + \tilde{\alpha}_v \delta P \\ &\quad + \sum_{j=h^*}^{t} i_d(v^*|_j)\left(\frac{Y_{t-j+u+1}}{i_d(v) - i_d(v^*|_u)} + \tilde{\alpha}_v Y_{t-j+1}\right),\end{aligned}$$

  all of which, except

$$\mathtt{SK}_{\varepsilon,i} + \frac{i_d(v^*|_u)Y_{t+1} - i_d(v)Y_{t+1}}{i_d(v) - i_d(v^*|_u)},$$

  are computable by B. The exception is computable as follows:

$$\begin{aligned}&\mathtt{SK}_{\varepsilon,i} + \frac{i_d(v^*|_u)Y_{t+1} - i_d(v)Y_{t+1}}{i_d(v) - i_d(v^*|_u)} \\ &= Y_{t+1} + \beta_i Y_1 - Y_{t+1} = \beta_i Y_1.\end{aligned}$$

  Therefore, B can compute $A_{v,i}$.
- Output $\mathtt{SK}_v = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,u}\}$ to A, where $\mathtt{sk}_{v,1} = \{A_{v,1}, B_v, C_{v,u+1}, \cdots, C_{v,t}\}$, $\mathtt{sk}_{v,i} = A_{v,i}$ $(i = 2, \cdots, u)$.

Challenge: When A decides Phase 1 is over, it outputs two messages $M_0, M_1 \in \mathbb{G}_2$ on which it wishes to be challenged. B picks a random bit $b \in \{0, 1\}$ and responds with the challenge ciphertext

$$C = \left(M_b \cdot T \cdot \hat{e}(Y_1, \gamma R), R, \left(\delta + \sum_{i=h^*}^{t} i_d(v^*|_i)\gamma_i\right)R\right),$$

where R and T are the input given to B. Note that $R = cP$ for $^{\exists}c \in \mathbb{Z}_q$ since $\mathbb{G}_1 = \langle P \rangle \ni R$

although c is unknown to B. Thus,

$$\left(\delta + \sum_{i=h^*}^{t} i_d(v^*|_i)\gamma_i\right) R$$
$$= c\left(\delta P + \sum_{i=h^*}^{t} i_d(v^*|_i)(Q_i + Y_{t-i+1})\right)$$
$$= c\left(\sum_{i=h^*}^{t} i_d(v^*|_i)Q_i + \delta P + \sum_{i=h^*}^{t} i_d(v^*|_i)Y_{t-i+1}\right)$$
$$= c\left(\sum_{i=h^*}^{t} i_d(v^*|_i)Q_i + P_1\right).$$

If $T = \hat{e}(P,R)^{\alpha^{t+1}}$, then C is a valid encryption of M_b under the ID v^* and the designated level h^*. Because $T \cdot \hat{e}(Y_1, \gamma R) = \hat{e}(P,R)^{\alpha^{t+1}}\hat{e}(P,\gamma R)^{\alpha} = \hat{e}(\alpha P, \alpha^t R + \gamma R) = \hat{e}(Q, R_{h^*})^c$. On the other hand, when T is uniform and independent in $\mathbb{G}_2$ (that is, it is sampled from $\mathcal{R}_{\mathtt{BDHE}}$), C is independent of b in the adversary's view.

Phase 2: A continues to issue queries not issued in Phase 1. B responds to them as above.

Guess: Finally, A outputs a guess $b' \in \{0,1\}$. Then B outputs a guess as follows. If $b = b'$, then B outputs 1, that is, it guesses $T = \hat{e}(P,R)^{\alpha^{t+1}}$. Otherwise, B outputs 0, that is, it guesses T is random in $\mathbb{G}_2$.

When the input T is sampled from $\mathcal{P}_{\mathtt{BDHE}}$, that is $T = \hat{e}(P,R)^{\alpha^{t+1}}$, then A's view is identical to its view in a real attack game and, therefore, A succeeds with the advantage of $|Pr[b = b'] - 1/2| \geq \varepsilon$. When the input T is sampled from $\mathcal{R}_{\mathtt{BDHE}}$, that is T is uniform in $\mathbb{G}_2$, then $Pr[b = b'] = 1/2$. Therefore, we arrive at

$$\left|\Pr\left[\begin{array}{l}(\mathbb{G}_1,\mathbb{G}_2,\hat{e}) \leftarrow \mathcal{IG}(1^k);\\ P,R \leftarrow \mathbb{G}_1; \alpha \leftarrow \mathbb{Z}_q :\\ A(\mathbb{G}_1,\mathbb{G}_2,\hat{e},P,R,Y_1,\cdots,Y_t,\\ \quad Y_{t+2},\cdots,Y_{2t+2},\hat{e}(P,R)^{\alpha^{t+1}}) = 0\end{array}\right]\right.$$
$$\left.-\Pr\left[\begin{array}{l}(\mathbb{G}_1,\mathbb{G}_2,\hat{e}) \leftarrow \mathcal{IG}(1^k);\\ P,R \leftarrow \mathbb{G}_1; \alpha \leftarrow \mathbb{Z}_q; T \leftarrow \mathbb{G}_2 :\\ A(\mathbb{G}_1,\mathbb{G}_2,\hat{e},P,R,Y_1,\cdots,Y_t,\\ \quad Y_{t+2},\cdots,Y_{2t+2},T) = 0\end{array}\right]\right|$$
$$\geq \left|\left(\frac{1}{2} \pm \varepsilon\right) - \frac{1}{2}\right| = \varepsilon.$$

It is easy to see that B simulates the environment of A perfectly. Thus, we have proved that AE-HIBE is `IND-sAE-HIBE-CPA` secure if $\mathcal{IG}$ satisfies the $(t+1)$-BDHE assumption. ∎

Remark: Our AE-HIBE is based on BBG-HIBE and, thus, the proof follows mostly that of BBG-HIBE. The important difference lies in the root secret key. In our AE-HIBE, each level, including the target level h^*, has each secret key. The root secret key consists of the t secret keys in total. On the other hand, there is one secret key in the root secret key in BBG-HIBE. In the set up phase of our proof, the secret information of Y_{t+1} has to be included into all components in the root secret key. If the secret information of Y_{t+1} is included into only a target-level-h^* component, which naturally corresponds to the proof on BBG-HIBE, the proof would fail.

In the challenge phase of our proof, the challenger makes ciphertext to v^* and the designated level h^*, where v^* might be appended random elements by B at the initialization phase. However, the attacker A cannot distinguish the cipher text to (his intention) v^* with the level h^* from that to (B's appended) v^* with the level h^* since the cipher text can be decrypted by $\mathtt{SK}_{v^*}|_h$ with (his intention) $v*$ and the cipher text itself is a random number. The same logic is used in the proof on BBG-HIBE.

Chosen Ciphertext Security. An efficient general method of constructing an IND-sID-CCA-secure HIBE from IND-sID-CPA-secure HIBE is proposed [4]. Applying the construction to our AE-HIBE results in IND-sAE-HIBE-CCA secure with the constant size ciphertext.

4.3 Efficiency

Table 2 summarizes the efficiency of AE-HIBE compared with BBG-HIBE. AE-HIBE realizes ID-based encryption with a hierarchical structure without any additional computation or memory cost, that is the performance of AE-HIBE is the same as that of BBG-HIBE. Therefore, AE-HIBE realizes the constant-size ciphertext and decryption time in the same way as BBG-HIBE and even reduces the encryption time of BBG-HIBE by designating ancestors with a level > 1 as receivers with a right of decryption.

5. Application

This section applies our AE-HIBE to the broadcast encryption. After briefly reviewing known facts on a public-key broadcast encryption, we reconsider a general model of subset-difference broadcast encryption and present a generic model based on AE-HIBE.

5.1 Known Facts on Broadcast Encryption

Broadcast encryption distributes digital contents to subscribers by using an open broadcast

Table 2 Efficiency of AE-HIBE and BBG-HIBE associated with a tree $\mathcal{T}$ with the level t.

	our AE-HIBE	BBG-HIBE [1)]
Key generation time	$O(t)$	$O(t)$
Public key size	$O(1)$	$O(1)$
Secret key size (at a node $v^{\dagger}$)	$O(t)$	$O(t)$
Key derivation time (at a node $v^{\dagger}$)	$O(t)$	$O(t)$
Encryption/Decryption time‡	$O(l-h)/O(1)$	$O(l)/O(1)$
Ciphertext length‡	$O(1)$	$O(1)$

†: the level of v is $l \leq t$.
‡: to a node v of level l and its designated ancestor of level h ($h \leq l$).

channel, in which a set of privileged users may be changed by each digital content. Let $\mathcal{N}$ be the set of all users with $|\mathcal{N}| = N$ and $\mathcal{R}$ be the set of all revoked users with $|\mathcal{R}| = r$. Then privileged users are in $\mathcal{N} \setminus \mathcal{R}$. The formal definition is given as follows.

Definition 6 (Broadcast encryption) Broadcast encryption consists of a 3-tuple of PPT algorithms (BE-Ini, BE-Enc, BE-Dec):

- $\texttt{BE-Ini}(1^k, t, \mathcal{N})$, the initialization algorithm, for input of the security parameter 1^k and a set of users $\mathcal{N}$ with $|\mathcal{N}| = N = 2^t$, outputs a system public parameter PK which includes two encryption algorithms E_1 for session keys and E_2 for messages, a family $\mathcal{S} = \{S_i\}$ of subsets of $\mathcal{N}$, the master's secret key SK which can compute all subset keys of $\mathcal{S}$, and a secret key K_u for a user $u \in \mathcal{N}$.
- $\texttt{BE-Enc}(\mathsf{PK}, \mathcal{N} \setminus \mathcal{R}, \mathcal{S}, K, M)$, the encryption algorithm, for input of the public parameter PK, a set of privileged users $\mathcal{N} \setminus \mathcal{R}$, a family $\mathcal{S}$ of sets, a session key K, and a message M, covers $\mathcal{N} \setminus \mathcal{R} = \cup_j S_{i_j}$ by disjoint subsets $\{S_{i_j}\}_j$ and encrypts the session key K and the message M with each subset key $K_{S_{i_j}}$ and K, respectively, to $C = \langle \{S_{i_j}\}_j, \{E_1(K_{S_{i_j}}, K)\}_j, E_2(K, M) \rangle$, where $E_i(K, M)$ $(i = 1, 2)$ means an encryption of M with a key K.
- $\texttt{BE-Dec}(\mathsf{PK}, K_u, C)$, for input of a secret key K_u of a user u, the public parameter PK, and a ciphertext C broadcasted by the center, finds a subset $S_{i_j} \ni u$, derives a subset key $K_{S_{i_j}}$ from K_u, decrypts $E_1(K_{S_{i_j}}, K)$ to K, and then decrypts $E_2(K, M)$ to M.

The efficiency of public-key broadcast encryption depends on the transmission rate $\sum_j |E_1(K_{S_{i_j}}, K)|$, the size of a user secret key $|K_u|$, and the size of user public keys. The number of subsets $|\mathcal{S}_u|$ that a user u belongs to relates to the number of a user secret key.

Two specific examples of subset-cover framework, the complete subtree broadcast encryption (CS) and the subset difference broadcast encryption (SD), are proposed [10)]. We focus on SD and improving the public-key SD. For the detailed explanation of CS and public-key CS, please refer to the original papers [6),10)]. In SD, users are assigned to leaves of a binary tree $\mathcal{T}$, where we assume that $|\mathcal{N}| = N = 2^t$ for the sake of simplicity. We call $\mathcal{T}$ the user tree. A privileged user is covered with a difference set of two subsets $S_{v|_h,v} = S_{v|_h} \setminus S_v$, where $v = v_1 \cdots v_h \cdots v_l$, its ancestor $v|_h = v_1 \cdots v_h$, and $S_{v|_h}$ or S_v is a complete subtree rooted at $v|_h$ or v, respectively. For a set $S_{v|_h,v}$, $v|_h$ or v is called primary root or secondary root [6)], respectively. We denote the path from $v|_h$ or the root of $\mathcal{T}$ to a user (leaf) u by $\rho_{v|_h,u}$ or ρ_u, respectively, and a set of nodes that just hang off $\rho_{v|_h,u}$ by $\mathcal{V}_{hang.\rho_{v|_h,u}}$. It is necessary to compress a user secret key K_u relative to $\mathcal{S}_u$ since $|\mathcal{S}_u| = O(N)$. For this purpose, the following two features on the *common* primary root and a privileged user are used for $v = v_1 \cdots v_h \cdots v_f \cdots v_l$, $w = v|_h$, and $v' = v|_f$):

- $S_{w,v} \supsetneqq S_{w,v'}$ (1)
- $S_{v|_h,v} \ni u$ (2)

$$\Leftrightarrow \begin{cases} \rho_u \ni v|_h, \\ \mathcal{V}_{hang.\rho_{v|_h,u}} \ni v|_f \text{ for } h <^{\exists} f \leq l \end{cases}$$

Eq. (1) means that a member in $S_{w,v'}$ is allowed to generate a subset key $K_{S_{w,v}}$ but a member in $S_{w,v} \setminus S_{w,v'}$ is not allowed to generate a subset key $K_{S_{w,v'}}$. This idea is used to compress a user key and derive a necessary subset key from the compression. The key derivation function, BE-KDer, is defined as follows:

•$\texttt{BE-KDer}(v|_h,\ v,\ \mathtt{SK}_{v|_h,v},\ \mathsf{PK}) \rightarrow (\mathtt{SK}_{v|_h,v1}, \mathtt{SK}_{v|_h,v0}, K_{S_{v|_h,v}})$:

For input of a primary root $v|_h$, a descendant node v, a secret key $\mathtt{SK}_{v|_h,v}$ of v to the primary

root $v|_h$, and public parameter PK, output secret keys $\mathtt{SK}_{v|_h,v0}$ and $\mathtt{SK}_{v|_h,v1}$ of v's children nodes $v0$ and $v1$ to the primary root $v|_h$, and a subset key $K_{S_{v|_h,v}}$, where $\mathtt{SK}_{w,v}$ means a secret key of a node v to a primary root w.
The initial secret key to the primary root w is set to $\mathtt{SK}_{w,w}$
$\mathtt{BE\text{-}KDer}(v|_h, v, \mathtt{SK}_{v|_h,v}, \mathsf{PK})$ satisfies the following properties for 3-tuple nodes $(v|_h, v|_f, v)$ of $v = v_1 \cdots v_h \cdots v_f \cdots v_l$:
• *One-way feature for a common primary root*: Given $\mathtt{SK}_{v|_h,v|_f}$ it is easy to compute $K_{S_{v|_h,v}}$, but given $\mathtt{SK}_{v|_h,v}$ it is difficult to compute $K_{S_{v|_h,v|_f}}$.
• It is difficult to compute $\mathtt{SK}_{v|_h,v}$ without knowledge of any secret key $\mathtt{SK}_{v|_h,v|_f}$ of an ancestor $v|_f$ of v.
A secret key of a user u is set to

$$K_u = \{\mathtt{SK}_{v,d^v}\}_{(v,d^v)\in\rho_u\times\mathcal{V}_{hang.\rho_{u,v}}},$$

which enables the user u to derive necessary subset key by using BE-KDer, which is assured by Eq. (2).

The public-key SD is realized by applying HIBE to the key derivation tree $\mathcal{T}_{\mathtt{KDT}}$ [6]. $\mathcal{T}_{\mathtt{KDT}}$ represents the inclusion relation in $\{S_{w,v}\}$ based on Eq. (1) and a subset key $\{K_{S_{w,v}}\}$. **Figures 7** and **8** present $\mathcal{T}$ and $\mathcal{T}_{\mathtt{KDT}}$ associated with $\mathcal{T}$ in the case of 4 users, respectively.

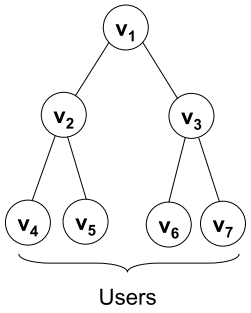

Fig. 7 The user tree $\mathcal{T}$ (4 users).

5.2 Reconsideration of Public-key SD Based on AE-HIBE

In order to reduce $|K_u|$, we further consider the next feature on a *common* secondary root:

$$\bullet\ S_{w,v} \supsetneqq S_{v',v}\ (v = v_1 \cdots v_h \cdots v_f \cdots v_l,\ w = v|_h,\ \text{and}\ v' = v|_f) \qquad (3)$$

Equation (3) dictates that a member in $S_{v',v}$ is allowed to generate a subset key $K_{S_{w,v}}$ but a member in $S_{w,v}\setminus S_{v',v}$ is not allowed to generate a subset key $K_{S_{v',v}}$. To make use of this feature, an additional condition is required for BE-KDer:
• *Ancestor-designated feature for a common secondary root*: For a designated ancestor $v|_h$ of $v = v_1 \cdots v_i \cdots v_h \cdots v_f \cdots v_l$, given $\mathtt{SK}_{v|_f,v}$ it is easy to compute $K_{S_{v|_h,v}}$, but given $\mathtt{SK}_{v|_i,v}$ it is difficult to compute $K_{S_{v|_h,v}}$.
The ancestor-excludable feature of AE-HIBE is perfectly suitable for this ancestor-designated feature for a common secondary root and can achieve the public-key SD more efficiently than the public-key based on HIBE. AE-HIBE does not need a key derivation tree $\mathcal{T}_{\mathtt{KDT}}$ associated with $\mathcal{T}$ and utilizes $\mathcal{T}$ to achieve the public-key SD.

5.3 Generic Construction of Public-key SD Based on AE-HIBE

We submit the generic construction of the public-key SD (BE-Ini, BE-Enc, BE-Dec) based on AE-HIBE (KGen, KDer, $\mathtt{KDer}_p$, Enc, Dec).
$\mathtt{BE\text{-}Ini}(1^k, t, \mathcal{N})$ executes the following:
(1) Run $\mathtt{KGen}(1^k, t)$ and get outputs of PK and the root secret key $\mathtt{SK}_\varepsilon$.
(2) Run $\mathtt{KDer}(\mathsf{PK}, v, \mathtt{SK}_\varepsilon)$ for $v = v_1 \cdots v_l$ from $l = 1$ to t one by one and generate node secret keys $\{\mathtt{SK}_v\}_{v\in\mathcal{T}}$.
(3) Set a user secret key K_u as a set of secret keys of nodes just hanging off the

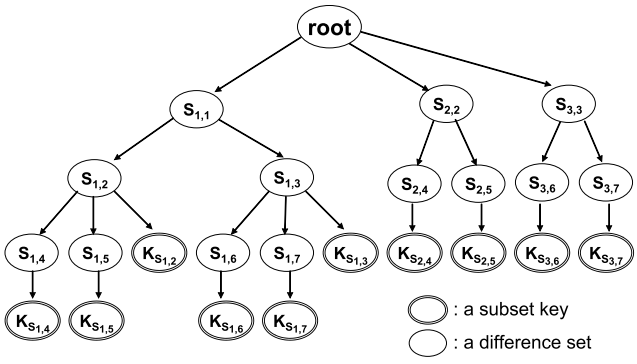

Fig. 8 The key derivation tree $\mathcal{T}_{\mathtt{KDT}}$.

Table 3 Comparison of generic models of SD and LSD.

	transmission rate	user-key size	public-key size	decryption time
Our public-key SD	$O(r\|C_{\mathtt{AE\text{-}HIBE}}\|)$	$O(\|\mathtt{SK}_{\mathtt{AE\text{-}HIBE}}\| \log N)$	$O(\|\mathsf{PK}_{\mathtt{AE\text{-}HIBE}}\|)$	$O(T_{\mathtt{BTE\text{-}KD}} \log N)$
public-key SD [6]	$O(r\|C_{\mathtt{HIBE}}\|)$	$O(\|\mathtt{SK}_{\mathtt{HIBE}}\| \log^2 N)$	$O(\|\mathsf{PK}_{\mathtt{HIBE}}\|)$	$O(T_{\mathtt{HIBE\text{-}KD}} \log N)$
public-key LSD [6]	$O(r\|C_{\mathtt{HIBE}}\|)$	$O(\|\mathtt{SK}_{\mathtt{HIBE}}\| \log^{3/2} N)$	$O(\|\mathsf{PK}_{\mathtt{HIBE}}\|)$	$O(T_{\mathtt{HIBE\text{-}KD}} \log N)$
SD [10]	$O(r)$	$O(\log^2 N)$	$--$	$O(\log N)$
LSD [8]	$O(r)$	$O(\log^{3/2} N)$	$--$	$O(\log N)$

$|C_{\mathtt{AE\text{-}HIBE}}|$ or $|C_{\mathtt{HIBE}}|$, $|\mathtt{SK}_{\mathtt{AE\text{-}HIBE}}|$ or $|\mathtt{SK}_{\mathtt{HIBE}}|$, $|\mathsf{PK}_{\mathtt{AE\text{-}HIBE}}|$ or $|\mathsf{PK}_{\mathtt{AE\text{-}HIBE}}|$, and $T_{\mathtt{BTE\text{-}KD}}$ or $T_{\mathtt{HIBE\text{-}KD}}$ present the ciphertext size, the node-secret-key size, the public-key size, and the key derivation time in AE-HIBE or HIBE, respectively.

path ρ_u, $\mathcal{V}_{hang.\rho_u}$, (there are $t+1$ nodes). Then $K_u = \{\mathtt{SK}_v\}_{v \in \mathcal{V}_{hang.\rho_u}}$.

(4) Output a system parameter PK, the SD-family $\mathcal{S} = \{S_{w,v}\}$, the master's secret key $\mathtt{SK}_\varepsilon$, and a secret key K_u for a user u together with an encryption algorithm E_2 for contents.

$\mathtt{BE\text{-}Enc}(\mathsf{PK}, \mathcal{N} \setminus \mathcal{R}, \mathcal{S}, K, M)$ executes the following:

(1) Cover $\mathcal{N} \setminus \mathcal{R} = \cup_v S_{v|_h,v}$ by disjoint subsets $S_{v|_h,v} \in \mathcal{S}$ in the same manner as SD, where $v = v_1 \cdots v_h \cdots v_l$.

(2) Encrypt K by $C_{v,h} \leftarrow \mathtt{Enc}(\mathsf{PK}, v, h, K)$ for each subset $S_{v|_h,v}$.

(3) Output $C = \langle \{S_{v|_h,v}\}_v, \{C_{v,h}\}_v, E_2(K, M) \rangle$.

$\mathtt{BE\text{-}Dec}(PK, K_u, C)$ executes the following:

(1) Find a subset $S_{v|_h,v} \ni u$ in the same manner as SD.

(2) Take a node secret key $\mathtt{SK}_{v|_f} \in K_u$, where $\mathtt{SK}_{v|_f} = \{\mathtt{sk}_{v|_f,1}, \cdots, \mathtt{sk}_{v|_f,h}, \cdots, \mathtt{sk}_{v|_f,f}\}$ with $v|_f \in \mathcal{V}_{hang.\rho_{v|_h,u}} \cap \rho_{v|_h,v}$ and $v = v_1 \cdots v_h \cdots v_f \cdots v_l$. Such $v|_f$ exactly exists from Eq. (2).

(3) Execute $\mathtt{KDer}_p(\mathsf{PK}, v|_f, \mathtt{SK}_{v|_f}, v|_{f+1})$ one by one to derive $\mathtt{SK}_v|_h = \{\mathtt{sk}_{v,1}, \cdots, \mathtt{sk}_{v,h}\}$ for $v = v_1 \cdots v_h \cdots v_f \cdots v_l$.

(4) Execute $\mathtt{Dec}(C, v, h, \mathtt{SK}_v|_h)$ to get K, decrypt $E_2(K, M)$ to M, and output M.

Table 3 compares performances of our construction and the previous constructions [6], which are the public-key CS, SD, and LSD [8]. The public-key LSD reduces the size of user keys of the public-key SD to $O(|\mathtt{SK}_{\mathtt{HIBE}}| \log^{3/2} N)$ with the double size of the transmission rate of the public-key SD, which still keeps the same order of $O(r|C_{\mathtt{HIBE}}|)$. Our novel ancestor-control feature of AE-HIBE can further reduce the size of user keys of the public-key LSD to $O(|\mathtt{SK}_{\mathtt{AE\text{-}HIBE}}| \log N)$ with half of the transmission rate. Table 1 compares concrete schemes of public-key broadcast encryptions, where our public-key SD uses our AE-HIBE; the public-key CS uses IBE [3]; and the public-key SD and LSD use BBG-HIBE. We see that our public-key broadcast encryption based on AE-HIBE works with the smallest user-key size $O(\log^2 N)$ while maintaining the constant public-key size and the transmission rate of $O(r)$.

6. Conclusion

We have proposed a new concept of ancestor-excludable hierarchical ID-based encryption (AE-HIBE), which can exclude ancestors with a level less than the designated one from a set of privileged ancestors. We have also given the functional definition together with the security definition. We have presented the concrete example of AE-HIBE, which can work with constant-size ciphertext and decryption time, independent of the hierarchy level, and is proven to be selective-ID secure in the standard model. Furthermore, AE-HIBE can naturally achieve the efficient public-key SD broadcast encryption (SD) with the user-key size of $O(\log^2 N)$ and the transmission rate of $O(r)$ for N users and r revoked users, which is the best performance between the public-key LSD and SD based on BBG-HIBE and the public-key SD based on our AE-HIBE.

Acknowledgments The author expresses our gratitude to anonymous referees for invaluable comments.

References

1) Boneh, D., Boyen, X. and Goh, E.: Hierarchical identity based encryption with constant size ciphertext, *Proc. Eurocrypt'05*, LNCS, Vol.3494, pp.440–456, Springer-Verlag (2005).

2) Boneh, D., Genrty, C. and Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys, *Proc. Crypto'05*,

LNCS, Vol.3621, pp.258–275, Springer-Verlag (2005).
3) Boneh, D. and Franklin, M.: Identity based encryption from the Weil pairing, *SIAM J. of Computing*, Vol.32, No.3, pp.586–615 (2003).
4) Boneh, D. and Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity based encryption, *Proc. CT-RSA'05*, Springer-Verlag (2005).
5) Canetti, R., Halevi, S. and Katz, J.: A forward-secure public-key encryption scheme, *Advances in Cryptology — Eurocrypt 2003*, LNCS, Vol.2656, Springer-Verlag (2003).
6) Dodis, Y. and Fazio, N.: Public Key Broadcast Encryption for Stateless Receivers, *Proc. ACM DRM '02*, LNCS, Vol.2696, pp.61–80, Springer-Verlag (2002).
7) Gentry, C. and Silverberg, A.: Hierarchical ID-Based Cryptography, *Asiacrypt 2002*, LNCS, Vol. 2501, pp.548–566, Springer-Verlag (2002).
8) Halevy, D. and Shamir, A.: The LSD broadcast encryption scheme, *Advances in Cryptology-Proceedings of Crypto'02*, LNCS, Vol.2442, pp.47–60, Springer-Verlag (2002).
9) Miyaji, A., Nakabayashi, M. and Takano, S.: New explicit conditions of Elliptic Curve Traces under FR-reduction, *IEICE Trans.*, Fundamentals, Vol.E84-A, No.5, pp.1234–1243 (2001).
10) Naor, D., Naor, M. and Lotspiech, J.: Revocation and Tracing Scheme for Stateless Receivers, *Advances in Cryptology-CRYPTO 2001*, LNCS, Vol.2139, pp.41–62, Springer-Verlag (2001).

Appendix

A.1 ID-based Encryption

This annex gives the formal definition of ID-based encryption (IBE) in the following. A concrete IBE is proposed in Ref. 3).

Definition 7 (IBE) IBE consists of a 4-tuple of PPT algorithms (`IBE-KGen`, `IBE-KDer`, `IBE-Enc`, `IBE-Dec`), where

- `IBE-KGen`(1^k), the center key generation algorithm, for input of the security parameter 1^k, outputs a system public key PK that includes the system parameter and the center's secret key (master-key) s.
- `IBE-KDer`(v, s), the center key derivation algorithm, for input of a user ID v and the center's secret key s, computes the user v's secret key $\mathtt{sk}_v$.
- `IBE-Enc`(PK, v, M), the encryption algorithm, for input of the public key PK, a user ID v, and a message M, computes a ciphertext C.
- `IBE-Dec`$(\mathsf{PK}, \mathtt{sk}_v, C) = M$, the decryption algorithm, for input of a user secret key $\mathtt{sk}_v$ and a ciphertext C, decrypts C to M.

A.2 Hierarchical ID-based Encryption

This annex gives the formal definition of HIBE in the following. A concrete HIBE is proposed in[1),7)].

Definition 8 (HIBE) HIBE consists of a 4-tuple of PPT algorithms[☆] (`HIBE-KGen`, `HIBE-KDer`, `HIBE-Enc`, `HIBE-Dec`), where

- `HIBE-KGen`$(1^k, t)$, the root center key-generation algorithm, for input of the security parameter 1^k and the level t of a tree, outputs a system public key PK that includes the system parameter and the root center's secret key $\mathtt{SK}_\varepsilon$.
- `HIBE-KDer`$(\mathsf{PK}, v, v_{l+1}, \mathtt{SK}_v)$, the key derivation algorithm, for input of the public key PK, a node $v = v_1 \cdots v_l$, a node secret key $\mathtt{SK}_v$, and a child vv_{t+1}, outputs vv_{l+1}'s secret key $\mathtt{SK}_{vv_{l+1}}$.
- `HIBE-Enc`(PK, v, M), the encryption algorithm, for input of the public key PK, a user ID $v = v_1 \cdots v_l$, and a message M, computes a ciphertext C.
- `HIBE-Dec`$(\mathtt{SK}_v, C, v)$, the decryption algorithm, for input of a user secret key $\mathtt{SK}_v$ and a ciphertext C, decrypts C to M.

(Received December 11, 2006)
(Accepted June 5, 2007)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.2, pp.000–000.)

☆ In Ref. 7) HIBE consists of 5 functions, that is, `HIBE-KDer` is separated into two algorithms: one chooses a secret random for generating a child's node key and the other computes the secret key itself. For simplicity, we combine them into one.

Atsuko Miyaji received the B. Sc., M. Sc., and Dr. Sci. degrees in mathematics from Osaka University, Osaka, Japan in 1988, 1990, and 1997 respectively. She joined Matsushita Electric Industrial Co., LTD from 1990 to 1998 and engaged in research and development for secure communication. She has been an associate professor at the Japan Advanced Institute of Science and Technology (JAIST) since 1998. She has joined the computer science department of the University of California, Davis since 2002. Her research interests include the application of number theory into cryptography and information security. She received the IPSJ Sakai Special Researcher Award in 2002, the Standardization Contribution Award in 2003, Engineering Sciences Society: Certificate of Appreciation in 2005, the AWARD for the contribution to CULTURE of SECURITY in 2007, and IPSJ/ITSCJ Project Editor Award in 2007. She is a member of the International Association for Cryptologic Research, the Institute of Electronics, Information and Communication Engineers, the Information Processing Society of Japan, and the Mathematical Society of Japan.