Journal of Information Processing Vol.23 No.5 562–569 (Sep. 2015)

[DOI: 10.2197/ipsjjip.23.562]

Regular Paper

Ciphertext Divided Anonymous HIBE and Its Transformation to Identity-Based Encryption with Keyword Search

Koji Tomida[1] Hiroshi Doi[2,a)] Masami Mohri[3,b)] Yoshiaki Shiraishi[4,c)]

Received: December 5, 2014, Accepted: June 5, 2015

Abstract: It has been shown that identity-based encryption with keyword search (IBEKS) can be constructed from level-2 anonymous hierarchical identity-based encryption (A-HIBE). A-HIBE is more complicated than non-anonymous HIBE (NA-HIBE). We have shown the definition of Ciphertext Divided A-HIBE (CD-A-HIBE). The basic idea of CD-A-HIBE is to make it possible to regard NA-HIBE as A-HIBE by dividing ciphertext into two parts so as not to leak the information of identity from the original ciphertext of NA-HIBE. We also have shown a concrete construction of CD-A-HIBE from BB1-HIBE, which is one of the NA-HIBE schemes and transformed BB1-CD-A-HIBE to IBEKS whose ciphertext is divided (CD-IBEKS). Then, the computational cost of CD-IBEKS is shown to be often more reasonable than that of IBEKS. In this paper, we show what type of NA-HIBE not limited to BB1-HIBE can be used for constructing CD-A-HIBE and how to transform a certain type of NA-HIBE to CD-A-HIBE generally. Then, we prove that these CD-A-HIBE schemes have indistinguishability and anonymity. The general transformation from CD-A-HIBE to CD-IBEKS is also shown. We prove that these CD-IBEKS schemes have indistinguishability.

Keywords: searchable encryption, public key encryption with keyword search, identity-based encryption with keyword search, indistinguishability, anonymity, hierarchical identity-based encryption

1. Introduction

Boneh et al. have proposed the first public key encryption with keyword search (PEKS) [3] which enables one to search for encrypted keywords without decryption. Boneh et al. also have proposed the transformation from an identity-based encryption (IBE) scheme to a PEKS scheme and shown the construction of PEKS based on the IBE scheme [4].

In IBE schemes, the receiver who can decrypt a ciphertext is specified by his identity, such as e-mail address, used in encryption. Abdalla et al. combined the concept of PEKS and IBE and have proposed the Identity-Based Encryption with Keyword Search (IBEKS) [1]. IBEKS is almost the same scheme as PEKS except for specifying the identity in encryption like IBE. In Ref. [1], IBEKS schemes are shown to be constructed from any level-2 anonymous HIBE schemes. Generally, the cost of an anonymous HIBE (A-HIBE) scheme is higher than that of a non-anonymous HIBE (NA-HIBE) scheme. Therefore, the cost of an IBEKS scheme is also high accordingly.

Searchable encryption schemes based on functional encryptions such as attribute-based encryption can perform fine-grained access control but the computational cost becomes high according to its complex construction. Although IBEKS can only perform simple access control, the computational cost should be low compared with other searchable encryption schemes because of its simple construction. Given this perspective, IBEKS should be suitable for applications which need only simple access control or low computational cost even if access control is not so functional. The computational cost of IBEKS should be lower to make use of the feature of low cost.

Ciphertext Divided A-HIBE (CD-A-HIBE) [6] for treating NA-HIBE as A-HIBE has been defined so as to reduce computational cost because it is known that IBEKS can be constructed from level-2 A-HIBE. In CD-A-HIBE, senders divide a ciphertext into two parts and send them to two servers respectively. A concrete construction of CD-A-HIBE from the BB1-HIBE scheme [2], which is one of the NA-HIBE schemes, is given by Ref. [6]. The definitions of the indistinguishability and the anonymity of CD-A-HIBE are also given and it was proven that BB1-CD-A-HIBE is anonymous [6]. Then, a Ciphertext Divided IBEKS (CD-IBEKS) scheme can be constructed from BB1-CD-A-HIBE. The computational cost of CD-IBEKS is often more reasonable than that of IBEKS in searching ciphertexts according to Ref. [6].

In this paper, we describe how to construct CD-A-HIBE not only from a BB1-HIBE scheme but also from other NA-HIBE schemes. We explain about HIBE in Section 2.2 and IBEKS constructed from level-2 A-HIBE in Section 2.3. In Section 3, the definitions of CD-A-HIBE and its security are described.

1 Nagoya Institute of Technology, Nagoya, Aichi 466–8555, Japan
2 Institute of Information Security, Yokohama, Kanagawa 221–0835, Japan
3 Gifu University, Gifu 501–1193, Japan
4 Kobe University, Kobe, Hyogo 657–8501, Japan
a) doi@iisec.ac.jp
b) mmohri@gifu-u.ac.jp
c) zenmei@port.kobe-u.ac.jp

In Section 4, we show what type of NA-HIBE schemes can be transformed to CD-A-HIBE schemes and give a security proof for the CD-A-HIBE schemes. In Section 5, the definitions of CD-IBEKS and its security are described. We show how to construct CD-IBEKS schemes from the CD-A-HIBE schemes generally. IBEKS schemes can be constructed from level-2 A-HIBE by hibe-2-ibeks transformation shown in Ref. [1], that is, we can construct CD-IBEKS schemes from CD-A-HIBE schemes in a similar way. We describe this transformation and give a security proof for the CD-IBEKS schemes in Section 6. Concrete constructions of CD-A-HIBE and CD-IBEKS from NA-HIBE which satisfies the conditions are described in Section 7.

2. Preliminary

2.1 Bilinear Groups

Let $\mathbb{G}_1$ be an additive group of prime order p. Let $\mathbb{G}_2$ be a multiplicative group of prime order p. Let P and Q be elements of $\mathbb{G}_1$. A paring $e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ has the following properties:

Bilinearity: For all $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, we have $e(aP, bQ) = e(P, Q)^{ab}$.

Non-degeneracy: $e(P, Q) \neq 1$ for all generator P, Q where 1 is a unit element in $\mathbb{G}_2$.

Computability: The function e can be efficiently computed.

2.2 Hierarchical Identity-Based Encryption (HIBE)

In an IBE scheme, a receiver contacts a third party, the Private Key Generator (PKG), to obtain his private key. The receiver must be authenticated by the PKG to obtain his private key from the PKG. There is only one PKG which can issue private keys to receivers.

On the other hand, hierarchical IBE (HIBE) is proposed in Ref. [5], where the root PKG is allowed to distribute the workload by delegating private key generation and identity authentication to lower-level PKGs.

In a HIBE scheme, a vector of strings $ID = (id_1, \cdots, id_l)$ represents an identity at depth l in the hierarchy. When $l = 0$, ID is the empty vector (). $ID|_{l-1} = (id_1, \ldots, id_{l-1})$ denotes the vector containing the first $l-1$ components of ID. Let $\mathrm{par}(ID) = ID|_{l-1}$ denote its parent. $usk[ID]$ is a private key corresponding to ID and H is a random oracle.

A HIBE scheme $\mathcal{HIBE}$ = (**Setup**, **KeyDer**, **Encrypt**, **Decrypt**) consists of the following four algorithms:

Setup: $(pk, msk = usk[()]) \xleftarrow{\$}$ **Setup**(1^k). $k \in \mathbb{N}$ is a security parameter. The root PKG generates public parameters pk and master secret key msk associated to the unique identity () at level 0.

KeyDer: $usk[ID] \xleftarrow{\$}$ **KeyDer**$^H(usk[par(ID)], ID)$. The private key for the identity ID is generated by his parent.

Encrypt: $C \xleftarrow{\$}$ **Encrypt**$^H(pk, ID, M)$. A sender encrypt a message M with an identity ID and obtain a ciphertext C.

Decrypt: $M \leftarrow$ **Decrypt**$^H(usk[ID], C)$. A receiver whose identity is ID decrypts ciphertext C to get a message.

In a HIBE scheme, the confidentiality of plaintext is proven by using the game of the indistinguishability (IND) of plaintexts as well as general public key cryptography.

Additionally, in A-HIBE and IBEKS, the confidentiality of identity (used as public key) is also required so that adversaries cannot know anything about identity used in encryption from the ciphertext. This property is called the anonymity and also proven by using the game of the anonymity (ANO). Refer to Ref. [1] for detailed definitions of IND (-CPA) and ANO (-CPA).

2.3 Identity-Based Encryption with Keyword Search (IBEKS)

A Public Key Encryption with Keyword Search (PEKS) scheme [3] can be constructed from an IBE scheme by regarding the receiver's identity of IBE as the keyword. In an IBE scheme, the ciphertext C of a plaintext M using the receiver's identity ID can be decrypted correctly by using the private key corresponding to the ID. Using this property, if the plaintext is decided in advance, whether the keyword included in the trapdoor (private key) is equal to the keyword used to generate the ciphertext or not can be tested in PEKS. Unlike IBE, the searcher cannot be specified by ID because ID is replaced with keyword w in a PEKS scheme.

Combining the concepts of IBE and PEKS, Abdalla et al. [1] proposed Identity-Based Encryption with Keyword Search (IBEKS). Compared with PEKS, in an IBEKS scheme, identity matching between identities included in trapdoor and ciphertext can be tested besides keyword matching. Therefore, a sender of a ciphertext can specify who can search the ciphertext by the identity specified in encryption. The IBEKS scheme $\mathcal{IBEKS}$ = (**Setup**, **KeyDer**, **Trapdoor**, **IBEKS**, **Test**) consists of the following five algorithms:

Setup: $(pk, msk = usk[()]) \xleftarrow{\$}$ **Setup**(1^k),

KeyDer: $usk[ID] \xleftarrow{\$}$ **KeyDer**$^H(msk, ID)$,

Trapdoor: $T_w \xleftarrow{\$}$ **Trapdoor**$^H(usk[ID], w)$,

IBEKS: $C \xleftarrow{\$}$ **IBEKS**$^H(pk, ID, w)$,

Test: $b \leftarrow$ **Test**$^H(T_w, C)$ $(b \in \{0, 1\})$.

For correctness, **Test**$^H(T_w, C)$ where $C =$ **IBEKS**$^H(pk, ID, w)$ as ciphertext and $T_{w'} =$ **Trapdoor**$^H(usk[ID'], w')$ as trapdoor, equals 1 meaning "accept" or "yes" if and only if both $w = w'$ and $ID = ID'$.

3. Ciphertext Divided A-HIBE (CD-A-HIBE)

An outline of Ciphertext Divided A-HIBE (CD-A-HIBE) proposed in Ref. [6] is as follows. The basic idea of CD-A-HIBE is to make it possible to regard NA-HIBE as A-HIBE by dividing a ciphertext into two parts so as not to leak the information of an identity from the original ciphertext of NA-HIBE.

In a CD-A-HIBE scheme, two divided ciphertexts are sent to two different servers. If an adversary gets both the divided ciphertexts, the adversary can obtain the original ciphertext and break the anonymity. To avoid the attack, two servers must not collude with each other. Furthermore, if the adversary can get both the divided ciphertexts from the communication between the sender and servers, the adversary can also break the anonymity. Therefore, a different public key encryption scheme from a HIBE scheme is used for secure communication.

3.1 Algorithms

CD-A-HIBE scheme $\mathcal{CD\text{-}A\text{-}HIBE}$ = (**Setup**, **KeyDer**,

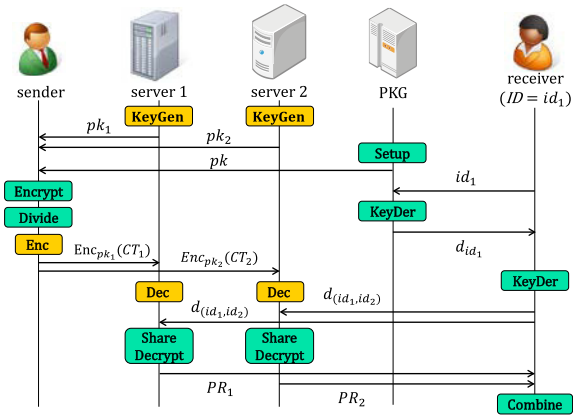

Fig. 1 The sequence of CD-A-HIBE.

Encrypt, **Divide**, **ShareDecrypt**, **Combine**) consists of the following six algorithms:

Setup: $(pk, msk) \xleftarrow{\$}$ **Setup**(1^k),

KeyDer: $usk[ID] \xleftarrow{\$}$ **KeyDer**$^H(usk[par(ID)], ID)$,

Encrypt: $CT \xleftarrow{\$}$ **Encrypt**$^H(pk, ID, M)$,

Divide: $(CT_1, CT_2) \xleftarrow{\$}$ **Divide**(CT),

ShareDecrypt:

$PR_1 \leftarrow$ **ShareDecrypt**$(usk[ID], CT_1)$,

$PR_2 \leftarrow$ **ShareDecrypt**$(usk[ID], CT_2)$,

Combine: $M \leftarrow$ **Combine**(PR_1, PR_2).

For correctness, the output of **Combine**(PR_1, PR_2), where $PR_1 =$ **ShareDecrypt**$(usk[ID], CT_1)$, $PR_2 =$ **ShareDecrypt**$(usk[ID], CT_2)$, $(CT_1, CT_2) =$ **Divide**(CT), $CT =$ **Encrypt**$^H(pk, ID, M)$ and $usk[ID] =$ **KeyDer**$^H(usk[par(ID)], ID)$, equals to the plaintext M.

3.2 Model

There are five entities in this model; sender, server1, server2, receiver and PKG. The processes of these entities in the sequence of level-2 CD-A-HIBE are shown as follows (see **Fig. 1**).

sender: A sender encrypts a message M with public parameters pk from PKG and an identity $ID = (id_1, id_2)$ of a receiver as public key. Then, the sender divides a ciphertext into CT_1 and CT_2, encrypts them over again with public keys pk_1 and pk_2 of server1 and server2 respectively and sends each ciphertext to each server.

server1, server2: Each server decrypts the ciphertext received from the sender with its own private key which corresponds to pk_1 (or pk_2) and obtains the divided ciphertext. After receiving the receiver's private key, each server decrypts the divided ciphertext partially. Each server sends the result of partial decryption to the receiver. Note that in this model, two servers do not collude with each other.

PKG: After receiving an identity id_1 from the receiver, PKG generates a private key d_{id_1} corresponding to id_1 and sends it to the receiver.

receiver: After receiving d_{id_1} from PKG, a receiver generates a private key $d_{(id_1,id_2)}$ corresponding to hierarchical identity $ID = (id_1, id_2)$ using d_{id_1}. Then, the receiver sends $d_{(id_1,id_2)}$ to each server as a partial decryption query. Then, the receiver receives two partial decryption results from two servers and combines them into the final decryption result.

Although the private key usually should not be given to a third party in public key cryptography, the private key is sent to each server because we give the transformation to CD-IBEKS.

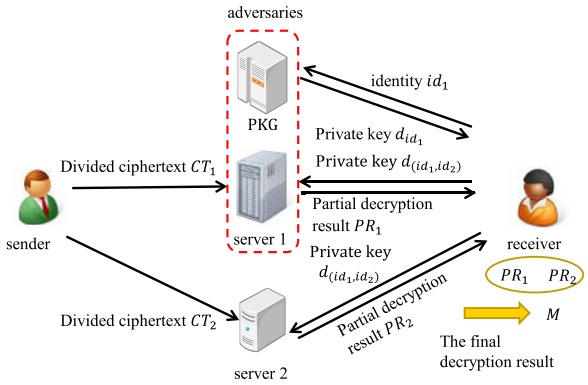

Fig. 2 Collusion of PKG.

3.3 Security

The indistinguishability and the anonymity of CD-A-HIBE are defined in Ref. [6] based on Ref. [1]. As shown in **Fig. 2**, it is supposed that one server (e.g., server1) colludes with PKG and that another server (e.g., server2) is honest but curious. If PKG colludes, the adversaries can generate any private keys. Therefore, a private key query in the security game is not needed. In the following explanation, let $MsgSp$ be the message space.

3.3.1 Indistinguishability (IND)

The indistinguishability of CD-A-HIBE is formalized as the experiment between adversary $\mathcal{A}$ and challenger C. $\mathcal{A}$ receives the divided ciphertext of $((id_1, id_2), M_0)$ or $((id_1, id_2), M_1)$ where id_1, id_2, M_0 and M_1 are selected by $\mathcal{A}$ and guesses which plaintext is encrypted.

Experiment $\mathbf{Exp}^{\text{CD-A-HIBE-IND-CPA-b}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$:

$(pk, msk) \xleftarrow{\$}$ **Setup**(1^k)

pick random oracle H

$((id_1, id_2), M_0, M_1, state) \xleftarrow{\$} \mathcal{A}^H(find, \{pk, msk\})$

If $|M_0| \neq |M_1|$ *or* $M_0, M_1 \notin MsgSp$ then return 0

$CT \xleftarrow{\$}$ **Encrypt**$(pk, (id_1, id_2), M_b)$

$(CT_1, CT_2) \xleftarrow{\$}$ **Divide**(CT)

$b' \xleftarrow{\$} \mathcal{A}^H(guess, CT_1, state)$

return b'.

$\mathcal{A}$ wins the game if $b = b'$. The advantage of $\mathcal{A}$ $\mathbf{Adv}^{\text{CD-A-HIBE-IND-CPA}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$ is defined as

$$\Pr\left[\mathbf{Exp}^{\text{CD-A-HIBE-IND-CPA-1}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{CD-A-HIBE-IND-CPA-0}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k) = 1\right].$$

CD-A-HIBE scheme $\mathcal{CD\text{-}A\text{-}HIBE}$ is said to be CD-A-HIBE-IND-CPA secure if the advantage is a negligible function in k for all polynomial-time adversaries $\mathcal{A}$.

3.3.2 Anonymity (ANO)

The anonymity of CD-A-HIBE is formalized as an experiment between adversary $\mathcal{A}$ and challenger C. $\mathcal{A}$ receives the divided ciphertext of $((id_{0,1}, id_{0,2}), M)$ or $((id_{1,1}, id_{1,2}), M)$ where $(id_{0,1}, id_{0,2})$, $(id_{1,1}, id_{1,2})$ and M are selected by $\mathcal{A}$ and guesses which hierarchical identity is used.

Experiment $\mathbf{Exp}^{\text{CD-A-HIBE-ANO-CPA-b}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$:

$(pk, msk) \xleftarrow{\$} \mathbf{Setup}(1^k)$

pick random oracle H

$((id_{0,1}, id_{0,2}), (id_{1,1}, id_{1,2}), M, state) \xleftarrow{\$} \mathrm{A}^H(find, \{pk, msk\})$

If $M \notin MsgSp$ then return 0

$CT \xleftarrow{\$} \mathbf{Encrypt}(pk, (id_{b,1}, id_{b,2}), M)$

$(CT_1, CT_2) \xleftarrow{\$} \mathbf{Divide}(CT)$

$b' \xleftarrow{\$} A^H(guess, CT_1, state)$

return b'.

$\mathcal{A}$ wins the game if $b = b'$. The advantage of $\mathcal{A}$ $\mathbf{Adv}^{\text{CD-A-HIBE-ANO-CPA}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$ is defined as

$$\Pr\left[\mathbf{Exp}^{\text{CD-A-HIBE-ANO-CPA-1}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{CD-A-HIBE-ANO-CPA-0}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k) = 1\right].$$

CD-A-HIBE scheme $\mathcal{CD\text{-}A\text{-}HIBE}$ is said to be CD-A-HIBE-ANO-CPA secure if the advantage is a negligible function in k for all polynomial-time adversaries $\mathcal{A}$.

4. Transformation from NA-HIBE to CD-A-HIBE

In this section, we focus on a level-2 NA-HIBE because we consider IBEKS as the main application. To describe the transformation to CD-A-HIBE from a certain kind of NA-HIBE, we firstly show the conditions on NA-HIBE in Section 4.1. Then, we describe the transformation and we provide the security proofs.

4.1 Conditions of NA-HIBE for Transformation to CD-A-HIBE

We show conditions on a level-2 NA-HIBE scheme which enables a transformation to CD-A-HIBE.

Setup:

There is no condition.

KeyDer:

The form of private key d_{ID} is limited as below.

$$d_{ID} = (d_1, \cdots, d_n) \in \mathbb{G}_1^n$$

Encrypt:

The form of ciphertext CT is limited as below.

$$CT = (A_1, \cdots, A_l, x_1, \cdots, x_n) \in \mathbb{G}_2^l \times \mathbb{G}_1^n$$

Decrypt:

The decryption algorithm using CT and d_{ID} is limited as below.

$$\prod_{i=1}^{l} A_i \frac{\prod_{i=j}^{n} e(x_i, d_i)}{\prod_{i=1}^{j-1} e(x_i, d_i)}$$

4.2 Transformation to CD-A-HIBE from NA-HIBE

We describe how to construct CD-A-HIBE from NA-HIBE which satisfies the conditions in Section 4.1.

Setup, **KeyDer**, **Encrypt**:

The algorithms of **Setup**, **KeyDer** and **Encrypt** are identical to the algorithms of NA-HIBE, respectively.

Divide:

To divide the ciphertext $CT = (A_1, \cdots, A_l, x_1, \cdots, x_n) \in \mathbb{G}_2^l \times \mathbb{G}_1^n$ into CT_1 and CT_2, all elements are divided into two parts. To divide $\mathrm{A}_i \in \mathbb{G}_2$, pick random elements $(a_{i,1}, a_{i,2}) \in \mathbb{Z}_p^2$, where $a_{i,1} + a_{i,2} = 1 \ (mod\ p)$ and output $A_{i,j} = A_i^{a_{i,j}}$ $(j = 1, 2)$. To divide $x_i \in \mathbb{G}_1$, pick random elements $(b_{i,1}, b_{i,2}) \in \mathbb{Z}_p^2$, where $b_{i,1} + b_{i,2} = 1 \ (mod\ p)$ and output $x_{i,j} = b_{i,j} x_i$ $(j = 1, 2)$. Thus, two divided ciphertexts are as follows: $CT_j = (A_{1,j}, \cdots, A_{l,j}, x_{1,j}, \cdots, x_{n,j})$ $(j = 1, 2)$.

ShareDecrypt:

To decrypt $CT_k = (A_{1,k}, \cdots, A_{l,k}, x_{1,k}, \cdots, x_{n,k})$ using the private key $d_{ID} = (d_1, \cdots, d_n)$, output $PR_k = \prod_{i=1}^{l} A_{i,k} \frac{\prod_{i=j}^{n} e(x_{i,k}, d_i)}{\prod_{i=1}^{j-1} e(x_{i,k}, d_i)}$ $(k = 1, 2)$ as a partial decryption result.

Combine:

To obtain the final decryption result, output $PR_1 \cdot PR_2$ using PR_1 and PR_2.

We can verify the correctness of the above algorithms as follows:

$$\begin{aligned}
&PR_1 \cdot PR_2 \\
&= \prod_{i=1}^{l} A_{i,1} \frac{\prod_{i=j}^{n} e(x_{i,1}, d_i)}{\prod_{i=1}^{j-1} e(x_{i,1}, d_i)} \prod_{i=1}^{l} A_{i,2} \frac{\prod_{i=j}^{n} e(x_{i,2}, d_i)}{\prod_{i=1}^{j-1} e(x_{i,2}, d_i)} \\
&= \prod_{i=1}^{l} A_{i,1} A_{i,2} \frac{\prod_{i=j}^{n} e(b_{i,1} x_i, d_i) e(b_{i,2} x_i, d_i)}{\prod_{i=1}^{j-1} e(b_{i,1} x_i, d_i) e(b_{i,2} x_i, d_i)} \\
&= \prod_{i=1}^{l} A_i^{a_{i,1}} A_i^{a_{i,2}} \frac{\prod_{i=j}^{n} e(x_i, d_i)^{b_{i,1}} e(x_i, d_i)^{b_{i,2}}}{\prod_{i=1}^{j-1} e(x_i, d_i)^{b_{i,1}} e(x_i, d_i)^{b_{i,2}}} \\
&= \prod_{i=1}^{l} A_i \frac{\prod_{i=j}^{n} e(x_i, d_i)}{\prod_{i=1}^{j-1} e(x_i, d_i)}
\end{aligned}$$

Note that $PR_1 \cdot PR_2$ is the same output of decrypt algorithm of NA-HIBE in Section 4.1.

4.3 Security

In Section 4.2, we designed the transformation from NA-HIBE which may use a random oracle. However, since a ciphertext is divided, IND and ANO can be proven without special simulation of the random oracle. We give a formal security proof as below.

4.3.1 Indistinguishability

We explain $\mathbf{Exp}^{\text{CD-A-HIBE-IND-CPA-b}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$ for CD-A-HIBE in Section 4.2. For a simple explanation, we assume that the server1 colludes with PKG.

Firstly, the simulation of the random oracle H is as follows.

(1) C maintains a list H^{list} of tuples $<q_i, ans_i>$ as explained below.

(2) If the query q_i already appears in H^{list}, then C responds with $H(q_i) = ans_i$.

(3) Otherwise, C generates ans_i randomly, adds the tuple $<q_i, ans_i>$ to the H^{list}, and responds with $H(q_i) = ans_i$.

If $b = 0$, $\mathbf{Exp}^{\text{CD-A-HIBE-IND-CPA-0}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$ is described as fol-

lows. The adversary $\mathcal{A}$ given pk and msk outputs $((id_1, id_2), M_0, M_1, state)$. The challenger C encrypts M_0 with pk and (id_1, id_2), generates $CT = (A_1, \cdots, A_l, x_1, \cdots, x_n)$. For $i = (1, \cdots, l)$, the challenger C picks random elements $(a_{i,1}, a_{i,2}) \in \mathbb{Z}_p^2$, where $a_{i,1} + a_{i,2} = 1$, and divides A_i into $A_{i,1}$ and $A_{i,2}$. For $i = (1, \cdots, n)$, the challenger C also picks random elements $(b_{i,1}, b_{i,2}) \in \mathbb{Z}_p^2$, where $b_{i,1} + b_{i,2} = 1$, and divides x_i into $x_{i,1}$ and $x_{i,2}$. Thus, C outputs $CT_1 = (A_{1,1}, \cdots, A_{l,1}, x_{1,1}, \cdots, x_{n,1}) \in \mathbb{G}_2^l \times \mathbb{G}_1^n$ and gives it to $\mathcal{A}$. Finally, $\mathcal{A}$ guesses which plaintext is encrypted M_0 or M_1 from CT_1 and outputs a guess $b' \in \{0, 1\}$.

If $b = 1$, $\mathbf{Exp}^{\text{CD-A-HIBE-IND-CPA-1}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$ is described similarly as above. Because both $\mathbb{G}_1$ and $\mathbb{G}_2$ are the groups of the same prime order p, CT_1 that is an output when $b = 0$ can be output when $b = 1$ by using appropriate random elements. Furthermore the probability that the CT_1 is output is $1/p^{l+n}$ because CT_1 is decided by $(l + n)$-tuple $(a_{1,1}, \cdots, a_{l,1}, b_{1,1}, \cdots, b_{n,1})$. This implies that CT_1 can be output with the same probability regardless of the case; case $b = 0$ or case $b = 1$. From this analysis, because even any computationally unbounded adversary cannot distinguish $b \in \{0, 1\}$, we can conclude $\mathbf{Adv}^{\text{CD-A-HIBE-IND-CPA}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k) = 0$ for all polynomial-time adversaries $\mathcal{A}$.

4.3.2 Anonymity

As described in Section 4.3.1, we can explain $\mathbf{Exp}^{\text{CD-A-HIBE-ANO-CPA-b}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$ for CD-A-HIBE in Section 4.2. For a simple explanation, we also assume that server1 colludes with PKG.

In the following experiment, the simulation of the random oracle H is the same as the simulation described in Section 4.3.1.

If $b = 0$, $\mathbf{Exp}^{\text{CD-A-HIBE-ANO-CPA-0}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k)$ is described as follows. The adversary $\mathcal{A}$ given pk and msk outputs $((id_{0,1}, id_{0,2}), (id_{1,1}, id_{1,2}), M, state)$. The challenger C encrypts M with $(id_{0,1}, id_{0,2})$, generates $CT = (A_1, \cdots, A_l, x_1, \cdots, x_n)$, picks random elements and divides CT into CT_1 and CT_2. The challenger C outputs $CT_1 = (A_{1,1}, \cdots, A_{l,1}, x_{1,1}, \cdots, x_{n,1})$. Finally $\mathcal{A}$ guesses which hierarchical identity is used for encryption and outputs a guess b'.

Using the same analysis shown in Section 4.3.1, we can conclude that CT_1 can be output with the same probability $1/p^{l+n}$ regardless of the case; case $b = 0$ or case $b = 1$. Because even any computationally unbounded adversary cannot distinguish $b \in \{0, 1\}$, we can conclude $\mathbf{Adv}^{\text{CD-A-HIBE-ANO-CPA}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{A}}(k) = 0$ for all polynomial-time adversaries $\mathcal{A}$.

5. Ciphertext Divided IBEKS (CD-IBEKS)

In this section, we explain algorithms of Ciphertext Divided IBEKS (CD-IBEKS) and its security. The model of CD-IBEKS is analogous to that of CD-A-HIBE in Section 3.2.

5.1 Algorithms

Ciphertext Divided IBEKS scheme $\mathcal{CD\text{-}IBEKS}$ = (**Setup**, **KeyDer**, **Trapdoor**, **CD-IBEKS**, **Divide**, **ShareSearch**, **ShareTest**) consists of the following seven algorithms:

Setup: $(pk, msk) \xleftarrow{\$}$ **Setup**(1^k),

KeyDer: $usk[ID] \xleftarrow{\$}$ **KeyDer**$^H(msk, ID)$,

Trapdoor: $T_w \xleftarrow{\$}$ **Trapdoor**$^H(usk[ID], w)$,

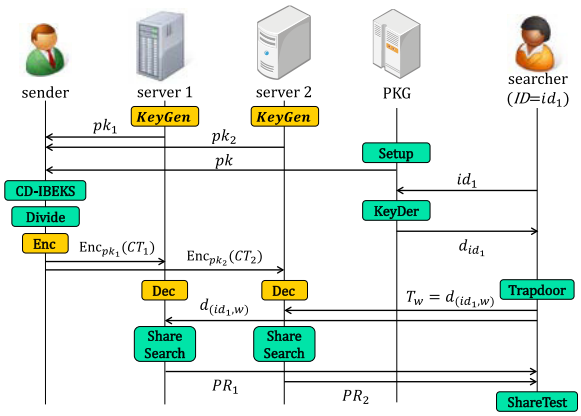

Fig. 3 The sequence of CD-IBEKS.

CD-IBEKS: $CT \xleftarrow{\$}$ CD-IBEKS$^H(pk, ID, w)$,

Divide: $(CT_1, CT_2) \xleftarrow{\$}$ **Divide**(CT),

ShareSearch: $PR_1 \leftarrow$ **ShareSearch**(T_w, CT_1),

$PR_2 \leftarrow$ **ShareSearch**(T_w, CT_2),

ShareTest: $b \leftarrow$ **ShareTest**(PR_1, PR_2).

For correctness, the output of **ShareTest**(PR_1, PR_2), where PR_1 = **ShareSearch**(T_w, CT_1), PR_2 = **ShareSearch**(T_w, CT_2), (CT_1, CT_2) = **Divide**(CT), CT = **CD-IBEKS**$^H(pk, ID, w)$, T_w = **Trapdoor**$^H(usk[ID], w)$ and $usk[ID]$ = **KeyDer**$^H(msk, ID)$, equals to 1, otherwise 0.

5.2 Model

There are five entities in this model; sender, server1, server2, searcher and PKG. The processes of these entities in the sequence of CD-IBEKS are shown as follows (see **Fig. 3**). A sender encrypts a keyword w with public parameters pk from PKG and an identity $ID = id_1$ of a searcher as public key. Then, the sender divides a ciphertext into two parts, encrypts them over again with each server's public key and sends each ciphertext to each server. After receiving the ciphertext, each server decrypts it with its own private key and obtains the divided ciphertext. A searcher generates a trapdoor T_W with a search keyword w and a private key d_{id_1} generated by PKG. The searcher sends a trapdoor $T_W = d_{(id_1,w)}$ to each server as a partial search query. After receiving the trapdoor, each server searches the divided ciphertext partially. Then, each server sends a partial search result to the searcher. The searcher combines them into the final search result.

5.3 Security

The information needed to be hidden is a keyword used in encryption in searchable encryption schemes. In CD-IBEKS, it is also needed to hide a search keyword from a ciphertext. Keyword indistinguishability of CD-IBEKS is formalized as the experiment between adversary $\mathcal{A}$ and challenger C. $\mathcal{A}$ receives the divided ciphertext of (ID, w_0) or (ID, w_1) where ID, w_0 and w_1 are selected by $\mathcal{A}$ and guesses which keyword is encrypted. It is supposed that one server (e.g., server1) colludes with PKG and that another server (e.g., server2) is honest but curious. If PKG colludes, the adversaries can generate any private keys and trapdoors. Therefore, a private key query and a trapdoor query in the security game are not needed. In the following explanation, let

$MsgSp$ be the message space.
Experiment $\mathbf{Exp}^{\text{CD-IBEKS-IND-CPA-b}}_{\mathcal{CD\text{-}IBEKS},\mathcal{A}}(k)$:

$(pk, msk) \xleftarrow{\$} \mathbf{Setup}(1^k)$
pick random oracle H
$(ID, w_0, w_1, state) \xleftarrow{\$} \mathcal{A}^H(find, \{pk, msk\})$
If $|M_0| \neq |M_1|$ *or* $M_0,\ M_1 \notin MsgSp$ then return 0
$CT \xleftarrow{\$} \mathbf{Encrypt}(pk, ID, w_b)$
$(CT_1, CT_2) \xleftarrow{\$} \mathbf{Divide}(CT)$
$b' \xleftarrow{\$} \mathcal{A}^H(guess, CT_1, state)$
return b'.

$\mathcal{A}$ wins the game if $b = b'$. The advantage of $\mathcal{A}$ $\mathbf{Adv}^{\text{CD-IBEKS-IND-CPA}}_{\mathcal{CD\text{-}IBEKS},\mathcal{A}}(k)$ is defined as

$$\Pr\left[\mathbf{Exp}^{\text{CD-IBEKS-IND-CPA-1}}_{\mathcal{CD\text{-}IBEKS},\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{CD-IBEKS-IND-CPA-0}}_{\mathcal{CD\text{-}IBEKS},\mathcal{A}}(k) = 1\right]$$

CD-IBEKS scheme $\mathcal{CD\text{-}IBEKS}$ is said to be CD-IBEKS-IND-CPA secure if the advantage is a negligible function in k for all polynomial-time adversaries $\mathcal{A}$.

6. Transformation from CD-A-HIBE to CD-IBEKS

We explain how to transform a CD-A-HIBE scheme to a CD-IBEKS scheme in Section 6.1. Then, we prove the security of the CD-IBEKS scheme transformed from the CD-A-HIBE scheme in Section 6.2.

6.1 Transformation to CD-IBEKS from CD-A-HIBE

The transformation from a level-2 A-HIBE scheme to an IBEKS scheme is shown in Ref. [1]. Encryption in IBEKS where id_1 is used as searcher's identity ID and w is used as a keyword is realized by encryption of A-HIBE using a hierarchical identity (id_1, w).

We describe how to construct CD-IBEKS from CD-A-HIBE. Given a CD-A-HIBE scheme $\mathcal{CD\text{-}A\text{-}HIBE}$ = (**Setup**, **KeyDer**, **Encrypt**, **Divide**, **ShareDecrypt**, **Combine**) with two levels, a transformation cd-a-hibe-2-cd-ibeks returns the CD-IBEKS scheme $\mathcal{CD\text{-}IBEKS}$ = (**Setup**, $\overline{\textbf{KeyDer}}$, **Trapdoor**, **CD-IBEKS**, $\overline{\textbf{Divide}}$, **ShareSearch**, **ShareTest**) as below.

Setup:
This is the same as **Setup** of CD-A-HIBE.

$\overline{\text{KeyDer}}$:
Taking the master secret key msk and the searcher's identity id_1 as input, output a private key $usk[id_1]$ using **KeyDer** of CD-A-HIBE.

Trapdoor:
When ID is id_1 and keyword is w, a hierarchical identity is (id_1, w). Using $usk[id_1]$ and w, compute T_w = $\mathbf{KeyDer}(usk[id_1], w)$.

CD-IBEKS:
When an ID is id_1 and keyword is w, hierarchical identity is (id_1, w). **CD-IBEKS**(pk, id_1, w) picks $R \xleftarrow{\$} MsgSp$, computes $CT \xleftarrow{\$} \mathbf{Encrypt}^H(pk, (id_1, w), R)$ and returns $\overline{CT} = (CT, R)$.

$\overline{\text{Divide}}$:
This is almost the same as **Divide**. $\overline{\textbf{Divide}}(CT)$ takes the ciphertext (CT, R), divides CT into $(CT_1, CT_2) \xleftarrow{\$} \mathbf{Divide}(CT)$ and returns $(\overline{CT_1}, CT_2) = ((CT_1, R), CT_2)$.

ShareSearch:
This is almost the same as **ShareDecrypt**. **ShareSearch**$(T_w, \overline{CT_1} = (CT_1, R)) = \overline{PR_1} = (PR_1, R)$ where PR_1 = **ShareDecrypt**(T_w, CT_1). On the other hand, **ShareSearch**$(T_w, CT_2) = PR_2$ where PR_2 = **ShareDecrypt**(T_w, CT_2).

ShareTest:
ShareTest$(\overline{PR_1} = (PR_1, R), PR_2)$ outputs 1 if and only if **Combine**$(PR_1, PR_2) = R$.

6.2 Security

We prove that if a CD-A-HIBE scheme is CD-A-HIBE-ANO-CPA secure, then a CD-IBEKS scheme via the transformation in Section 6.1 is CD-IBEKS-IND-CPA secure. Since a ciphertext is divided, no special simulation is needed for a random oracle as in Section 4.3.

Theorem 1
Let $\mathcal{CD\text{-}A\text{-}HIBE}$ be a CD-A-HIBE scheme and let $\mathcal{CD\text{-}IBEKS}$ = cd-a-hibe-2-cd-ibeks($\mathcal{CD\text{-}A\text{-}HIBE}$). If $\mathcal{CD\text{-}A\text{-}HIBE}$ is CD-A-HIBE-ANO-CPA secure, then $\mathcal{CD\text{-}IBEKS}$ is CD-IBEKS-IND-CPA secure.

Proof:
Suppose that $\mathcal{A}$ is an adversary that breaks CD-IBEKS-IND-CPA security, $\mathcal{B}$ is an adversary that breaks CD-A-HIBE-ANO-CPA security and C is a simulator of CD-A-HIBE. We will show how to use $\mathcal{A}$ in the construction of an adversary $\mathcal{B}$. For a simple explanation, we assume that server 1 colludes with PKG. The game among $\mathcal{A}$, $\mathcal{B}$ and C is as follows.

Setup:
The simulator C executes **Setup** of CD-A-HIBE and gives public parameter pk and master secret key msk to $\mathcal{B}$. $\mathcal{B}$ gives them to $\mathcal{A}$.

The simulation of a random oracle is identical to the simulation described in Section 4.3.

Challenge:
The adversary $\mathcal{A}$ sends challenge ID ID^* and two challenge keywords w_0^* and w_1^* to $\mathcal{B}$. $\mathcal{B}$ picks a challenge message R^* from a message space randomly and sends $(R^*, ID_0^* = (ID^*, w_0^*)$, $ID_1^* = (ID^*, w_1^*))$ to C. C picks a random bit $b \xleftarrow{\$} \{0, 1\}$ and generates a ciphertext CT^* to encrypt R^* with ID_b^*. Then, C divides CT^* into CT_1^* and CT_2^* and sends (R^*, CT_1^*) to $\mathcal{B}$. $\mathcal{B}$ forwards it to $\mathcal{A}$.

Guess:
$\mathcal{A}$ outputs its guess b' and sends it to $\mathcal{B}$. $\mathcal{B}$ forwards it to C as its own output.

The above simulation is perfect. Furthermore, $\mathcal{B}$ wins the game whenever $\mathcal{A}$ does. Therefore, we have that $\mathbf{Adv}^{\text{CD-IBEKS-IND-CPA}}_{\mathcal{CD\text{-}IBEKS},\mathcal{A}}(k) \leq \mathbf{Adv}^{\text{CD-A-HIBE-ANO-CPA}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{B}}(k)$. Since we have shown that $\mathbf{Adv}^{\text{CD-A-HIBE-ANO-CPA}}_{\mathcal{CD\text{-}A\text{-}HIBE},\mathcal{B}}(k) = 0$ in Section 4.3.2, we can conclude $\mathbf{Adv}^{\text{CD-IBEKS-IND-CPA}}_{\mathcal{CD\text{-}IBEKS},\mathcal{A}}(k) = 0$. If in the case that the server2 colludes with PKG and the server1 is honest but cu-

rious, we can prove in the same way. The difference is just that the ciphertext CT_2 does not include a randomly selected R^*. This completes the proof of the theorem.

7. Concrete Construction

In this section, we show a concrete construction of CD-A-HIBE scheme based on an NA-HIBE scheme which satisfies the conditions described in Section 4.1.

7.1 BBG05-HIBE

An example of an NA-HIBE is a BBG05-HIBE scheme [7]. A construction of BBG05-HIBE limited to level-2 hierarchy is as below.

Setup:

This is the same algorithm described in Ref. [7] limited to level-2 hierarchy as follows. The public parameters pk are $(P, P_1, P_2, P_3, h_1, h_2) \in \mathbb{G}_1^6$ and the master secret key msk is $\alpha P_2 \in \mathbb{G}_1$. Here, $\alpha \in \mathbb{Z}_p^*$ is an element picked at random.

KeyDer:

The private key d_{ID} for an identity $ID = (id_1, id_2)$ is $(\alpha P_2 + r(id_1 h_1 + id_2 h_2 + P_3), rP) \in \mathbb{G}_1^2$. Here, $r \in \mathbb{Z}_p$ is a random element.

Encrypt:

To encrypt a message $M \in \mathbb{G}_2$ under the public key $ID = (id_1, id_2) \in \mathbb{Z}_p^2$, pick a random $s \in \mathbb{Z}_p$ and output $CT = (e(P_1, P_2)^s \cdot M, sP, s(id_1 h_1 + id_2 h_2 + P_3)) \in \mathbb{G}_2 \times \mathbb{G}_1^2$.

Decrypt:

To decrypt a given ciphertext $C = (A_1, x_1, x_2)$ using the private key $d_{ID} = (d_1, d_2)$, output $M' = A_1 \cdot \frac{e(x_2, d_2)}{e(x_1, d_1)}$.

7.2 BBG05-CD-A-HIBE

From the description of Section 7.1, the algorithms of **KeyDer**, **Encrypt** and **Decrypt** of BBG05-HIBE satisfy the conditions described in Section 4.1. The algorithms of **Setup**, **KeyDer** and **Encrypt** of BBG05-CD-A-HIBE are the same algorithms as those of BBG05-HIBE. If the algorithms of **Divide**, **ShareDecrypt** and **Combine** are given as below, then a tuple of these algorithms (**Setup**, **KeyDer**, **Decrypt**, **Divide**, **ShareDecypt**, **Combine**) become a CD-A-HIBE scheme satisfying IND and ANO.

Divide:

The ciphertext CT is (A_1, x_1, x_2). Pick random elements $(a_{1,1}, a_{1,2}) \in \mathbb{Z}_p^2$ such as $a_{1,1} + a_{1,2} = 1 \ (mod\ p)$, and divide $A_1 \in \mathbb{G}_2$ into $A_{1,j} = A_1^{a_{1,j}}$ $(j = 1, 2)$. Pick also random elements $(b_{1,1}, b_{1,2}) \in \mathbb{Z}_p^2$ $(i = 1, 2)$ such as $b_{i,1} + b_{i,2} = 1 \ (mod\ p)$, and divide $x_i \in \mathbb{G}_1$ into $x_{i,j} = b_{i,j} x_i$ $(x_i \in \mathbb{G}_1)$ $(i = 1, 2, j = 1, 2)$. Then, $CT_j = (A_{1,j}, x_{1,j}, x_{2,j})$ $(j = 1, 2)$ are output.

ShareDecrypt:

To decrypt $CT_i = (A_1, x_{1,i}, x_{2,i})$ using the private key $d_{ID} = (d_1, d_2)$, output $PR_i = A_i \cdot \frac{e(x_{2,i}, d_2)}{e(x_{1,i}, d_1)}$ as a partial decryption result.

Combine:

The output of **Combine**(PR_1, PR_2) equals to $PR_1 \cdot PR_2$. We can verify the correctness of the above algorithms as follows: $PR_1 \cdot PR_2 = A_1^{a_{1,1}} \cdot \frac{e(b_{2,1} x_2, d_2)}{e(b_{1,1} x_1, d_1)} \cdot A_1^{a_{1,2}} \cdot \frac{e(b_{2,2} x_2, d_2)}{e(b_{1,2} x_1, d_1)} = A_1 \cdot \frac{e(x_2, d_2)}{e(c_1, d_1)}$. Note that $A_1 \cdot \frac{e(x_2, d_2)}{e(c_1, d_1)}$ is the same output of BBG05-HIBE Decrypt algorithm.

We can also obtain a construction of CD-IBEKS from this CD-A-HIBE by using the transformation in Section 6.1. Any other NA-HIBE schemes which satisfy the condition can also be transformed to CD-A-HIBE and CD-IBEKS in this way.

8. Conclusion

In this paper, we showed the conditions of NA-HIBE which enable transformation to CD-A-HIBE. Then, we showed how to transform a certain kind of NA-HIBE scheme to CD-A-HIBE generally. We gave the proof that CD-A-HIBE schemes transformed from a certain kind of NA-HIBE schemes have indistinguishability and anonymity.

We also showed the transformation of a CD-A-HIBE scheme to a CD-IBEKS scheme. This enables the construction of a CD-IBEKS scheme based not on A-HIBE schemes but NA-HIBE schemes for computational cost reduction. We gave the proof that CD-IBEKS schemes transformed from CD-A-HIBE schemes have indistinguishability.

Acknowledgments This work was supported by JSPS KAKENHI Grant Numbers 25330151, 25330161.

References

[1] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P. and Shi, H.: Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions, *CRYPTO 2005*, *LNCS*, Vol.3621, pp.205–222 (2005).

[2] Boneh, D. and Boyen, X.: Efficient Selective Identity-Based Encryption Without Random Oracles, *EUROCRYPT 2004*, *LNCS*, Vol.3027, pp.223–238 (2004).

[3] Boneh, D., Crescenzo, G.D., Ostrovsky, R. and Persiano, G.: Public Key Encryption with Keyword Search, *EUROCRYPT 2004*, *LNCS*, Vol.3027, pp.506–522 (2004).

[4] Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing, *CRYPTO 2001*, *LNCS*, Vol.2139, pp.213–229 (2001).

[5] Gentry, C. and Silverberg, A.: Hierarchical ID-Based Cryptography, *ASIACRYPT 2002*, *LNCS*, Vol.2501, pp.548–566 (2002).

[6] Tomida, K., Doi, H., Mohri, M. and Shiraishi, Y.: Construction of Ciphertext Divided Identity-Based Encryption with Keyword Search, *Proc. Computer Security Symposium 2014* (*CSS 2014*), Vol.2014, No.2, pp.551–558, Information Processing Society of Japan (2014).

[7] Boneh, D., Boyen, X. and Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext, *EUROCRYPT 2005*, *LNCS*, Vol.3494, pp.440–456 (2005).

Koji Tomida received B.E. degree from Nagoya Institute of Technology, Japan, in 2013. He is a master course student of the institute. His current research interests include information security and cryptography.

Hiroshi Doi received B.S. degree in mathematics from Okayama University in 1988, M.S. degree in information science from JAIST in 1994, and D.S. degree from Okayama University in 2000, respectively. He is currently a professor at the Graduate School of Information Security, Institute of Information Security, Japan. His research interests include information security and cryptography.

Masami Mohri received B.E. and M.E. degrees from Ehime University, Japan, in 1993 and 1995 respectively. She received Ph.D. degree in Engineering from the University of Tokushima, Japan in 2002. From 1995 to 1998 she was an assistant professor at the Department of Management and Information Science, Kagawa junior college, Japan. From 1998 to 2002 she was a research associate at the Department of Information Science and Intelligent Systems, University of Tokushima, Japan. From 2003 to 2008 she was a lecturer of the same department. Since 2008, she has been an associate professor at the Information and Multimedia Center, Gifu University, Japan. Her research interests are in coding theory, information security and cryptography. She is a member of IEEE and a senior member of IEICE.

Yoshiaki Shiraishi received B.E. and M.E. degrees from Ehime University, Japan, and Ph.D. degree from University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a lecturer at the Department of Informatics, Kinki University, Japan. From 2006 to 2013 he was an associate professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an associate professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and knowledge sharing and creation support. He received the SCIS 20th Aniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. He is a member of IEEE, ACM and a senior member of IEICE.