Expressive Ciphertext-Policy Attribute-Based Encryption with Fast Decryption

Hikaru Tsuchida[1]*, Takashi Nishide[2], and Eiji Okamoto[2]
[1]NEC Corporation, Kanagawa, 211-8666, Japan
h-tsuchida@bk.jp.nec.com
[2]Faculty of Engineering, Information and Systems, University of Tsukuba, Ibaraki, 305-8573, Japan
{nishide,okamoto}@risk.tsukuba.ac.jp

Abstract

Attribute-Based Encryption (ABE) is a cryptosystem which supplies access control for an encrypted data in a cloud storage and has been actively studied. However, in the ABE system, a receiver needs much time to decrypt an encrypted data. It is because the cost of pairing operations for decryption becomes heavy linearly with the size of an access structure specified for ciphertexts. Due to this, the construction of ABE is required to reduce the number of pairing operations and achieve expressiveness of an access structure simultaneously. In this paper, we propose a new construction of ciphertext-policy ABE supporting general predicates with a constant number of pairing operations for decryption. We also prove that our construction achieves new security notion which we introduce, restricted-selectively payload-hiding security under the q-type decisional bilinear Deiffie-Hellman assumption.

Keywords: Attribute-Based Encryption, Non-monotone Access Structure, Fast Decryption

1 Introduction

1.1 Background

A cloud storage service has attracted a lot of attension recently. It is the outsourcing of data storage to cloud service providers. Cloud storage services have properties of cost savings and flexibility, but those do not seem to be secure. For example, if the cloud storage server is compromised, there may be leaks of sensitive data.

Attribute-Based Encryption (ABE) [23, 13, 10, 21, 9, 27, 14, 16] is considered as one of the best methods for access control of data which is stored in the cloud storage server. ABE is the public key cryptosystem which supplies data security and access control without needing to trust the cloud and it enables to specify access policies and associated attributes among encrypted data and users' private keys. ABE is roughly divided into two types: Key-Policy ABE (KP-ABE) [13] (which specifies access policies to users' private keys and associated attributes to ciphertexts) and Ciphertext-Policy ABE (CP-ABE) [10] (which specifies access policies to ciphertexts and associated attributes to users' private keys). For example, in the CP-ABE system, there are three entities: Key Generation Center (KGC), user, data holder. The KGC publishes public keys related to attributes for encryption. Each user (who receives an encrypted data and tries to decrypt it) has their own private keys related to his/her attributes which are issued by the KGC. A data holder gets public keys from the KGC and encrypts his/her data with an access policy, and stores it in a cloud storage. A typical example is as follows: If an employee (who is

Journal of Internet Services and Information Security (JISIS), volume: 8, number: 4 (November 2018), pp. 37-56

*Corresponding author: NEC Corporation, 1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa, 211-8666, Japan, Tel: +81-044-431-7686. The major part of this work was completed while the corresponding author was a graduate student at University of Tsukuba.

the data holder) encrypts an internal document with an access policy "sales department AND manager" and stores it in the external company server, only the sales manager (who has the private keys related to "sales dapartment" and "manager") is able to decypt the encrypted internal document. Even if the external company server is compromised, there is no leak of the internal document because their internal document is encrypted. Also ABE is already mature enough to be deployed in applications to IoT devices [7, 19, 6, 25].

Most existing ABE constructions are pairing-based ones [23, 13, 10, 21, 9, 27, 14, 16, 20]. Roughly speaking, a pairing function is a polynomial-time computable nondegenrate bilinear map. These constructions have the drawback in that the number of pairing operations for decryption is proportional to the number of attributes which are used to specify an access policy. For example, there exists several pairing-based previous works as follow: Hohenberger and Waters proposed a selectively secure CP-ABE scheme supporting monotone access structures with only 2 pairing operations for decryption in Section 3.5 of [16]. Malluhi et al. proposed a monotone and selectively secure CP-ABE with the same number of pairing operations but shorter ciphertext than [16] in [18]. Agrawal and Chase proposed monotone and adaptively secure CP-ABE with 6 pairing operations for decryption in [4]. Zhang et al. proposed a selectively secure CP-ABE scheme supporting AND predicates (multiple values and wildcard) and achieving short size ciphertexts with only 2 pairing operations for decryption [27].

On the other hand, there are previous works aiming to improve expressiveness of access structures. Ostrovsky et al. proposed a KP-ABE scheme which supports non-monotone access structures in [21]. Okamoto and Takashima proposed an adaptively secure CP-ABE scheme supporting non-monotone access structures with inner-product relations [20]. The non-monotone access structure is desirable in terms of efficiency of attribute management and ciphertext size. For example, we assume that there are attribute categories t_1 and t_2 in an ABE system. t_1 includes attributes (or attribute vectors) a_1, b_1, c_1 and d_1. t_2 includes attributes (or attribute vectors) a_2 and b_2. Here we consider access policy $\neg a_1 \wedge a_2$. If the ABE system is supprts only the monotone access structure, the KGC needs to manage $\neg a_1$ additionally or the ciphertext size becames larger except the constant-size ABE system because the access policy must be expressed as $(b_1 \vee c_1 \vee d_1) \wedge a_2$. If the ABE system can support a non-monotone access structure, this system can express the access policy straightforwardly, so there is no additional cost. Hence, the ABE system supprting a non-monotone access structure is superior to the one supprting only the monotone access structure. Especially, the CP-ABE scheme proposed by Okamoto and Takashima [20] can support non-monotone access structures with inner-product relations, i.e., it can suppport more expressive access structure than other ABE systems supporting non-monotone one. However, the CP-ABE scheme in [20] needs a number of pairing operations which are proportional to the number of attributes and its dimensions satisfying an access policy.

For a device which has only low computation resources, many pairing operations lead to a very heavy task. Therefore, the CP-ABE scheme supporting expressive access structures (as [20]) with a constant number of pairing operations for decryption (as [16, 18, 27, 4]) is more desirable.

1.2 Our Results

We propose a new CP-ABE scheme supporting non-monotone access structures with inner-product relations with three pairing operations for decryption. There is no CP-ABE scheme that supports non-monotone access structures with constant pairing operations except our proposal as far as we know. We also introduce a new security model, restricted-selectively payload-hiding (r-PH) security against the chosen plaintext attacks that is weaker than selectively payload-hiding security against the chosen plaintext attacks in [26]. We prove that our construction achieves r-PH security under the q-type decisional bilinear Diffie-Hellman assumption.

We note that our construction can be viewed as an extension of [16, Section 3.5] obtained by com-

bining and adapting the techniques of Functional Encryption for Inner Products (FEIP) due to Abdalla et al. [1] and general access structures due to Okamoto and Takashima [20] to [16, Section 3.5]. Therefore, if an attribute category has only one attribute and an access structure is monotone, our construction achieves selectively payload-hiding security against the chosen plaintext attacks in [26]. If an attribute category has some attributes and/or an access structure is non-monotone (i.e., the same attribute setting and access structure as [20]), our construction achieves r-PH security against the chosen plaintext attacks.

That is, our construction has a tradeoff with [20] in that our scheme is restricted-selectively secure but needs only a constant number of pairing operations. Hence even if some users own devices which have only low computation resources, they can be utilized with our construction.

1.3 Key Techniques

The proposed scheme is composed by conjunction of the FEIP proposed by Abdalla et al. [1] and the selectively secure CP-ABE supporting monotone access structures with constant pairing operations for decryption proposed by Hohenberger and Waters [16]. Our design of the access structure is based on [20] proposed by Okamoto and Takashima. In [20], the access structure is a non-monotone access structure with inner-product relations which is realized by combining monotone span programs and inner-product predicates. The scheme [20] uses the property of Dual Pairing Vector Spaces (DPVS) to express the inner-product predicate, but we use a modified FEIP to specify the inner-product predicates. We also employ the selectively secure CP-ABE with fast decryption [16] for realizing monotone span programs. In this way, our proposed scheme realizes the same access control as [20] (which is more expressive than [16]) and a constant number of pairing operations.

1.4 Related Work

1.4.1 ABE

The notion of Fuzzy Identity-Based Encryption (FIBE) was introduced by Sahai and Waters [23]. It was a special type of ABE. FIBE supports only the threshold access structure. After FIBE [23], KP-ABE [13] was introduced by Goyal et al. Then, CP-ABE [10] was introduced by Bethencourt et al. These schemes [23, 10, 13] support only the monotone access structure, but Ostrovsky et al. proposed a scheme which supports non-monotone access structures in [21]. Waters proposed a scheme which achieves selective security based on the standard assumption in the standard model and supports monotone access structures by using Linear Secret Sharing Scheme (LSSS) in [26]. Okamoto and Takashima proposed an adaptively secure scheme which is built in prime-order groups and supports non-monotone access structures with inner-product relations by introducing DPVS in [20].

In the KP-ABE area, Attrapadung et al. proposed non-monotone and selectively secure KP-ABE with constant-size ciphertexts [9]. In [8], Attrapadung proposed monotone and adaptively secure KP-ABE with constant-size ciphertexts in composite-order groups. Takashima proposed non-monotone and semi-adaptively secure KP-ABE with constant-size ciphertexts in prime-order groups [24]. In [3], Agrawal and Chase proposed monotone and semi-adaptively secure KP-ABE with shorter ciphertexts than [24] in prime-order groups. Kim et al. proposd monotone and semi-adaptively secure KP-ABE with shorter ciphertext than [3] in [17]. Constant-size ciphertexts lead to a constant number of pairing operations. Hohenberger and Waters proposed monotone KP-ABE with fast decryption [16].

In the CP-ABE area, Hohenberger and Waters proposed a monotone and selectively secure CP-ABE scheme with fast decryption (which is modified CP-ABE in [26, Section 5]) in [16, Section 3.5]. It needs 2 pairing operations for decryption. Malluhi et al. [18] proposed a monotone and selectively secure CP-ABE scheem with 2 pairing operations and shorter ciphertext than [16]. Agrawal and Chase

[4] proposed a monotone and adaptively secure CP-ABE scheme with 6 times pairing computation for decryption. In [27], Zhang et al. proposed a CP-ABE scheme with constant-size ciphertext. In [22], Rao and Dutta proposed a CP-ABE scheme with fast decryption and multiple authorities without central authority. Green and Hohenberger proposed a monotone CP-ABE scheme with outsourcing part of decryption in [14].

1.4.2 FEIP

Abdalla et al. proposed FEIP [1]. FEIP is the functional encryption for inner-prodcut functionality. In FEIP, if a user who has a private key which is associated with a vector $\vec{x}$ decrypts an encrypted vector $\vec{y}$, the user gets inner product value $\vec{x}\cdot\vec{y}$ and nothing else. In [1], Abdalla et al. proposed a generic scheme and instantiations whose selective security is based on Decision Diffie-Hellman assumption (DDH) and Learning With Errors assumption (LWE). In [5], Agrawal et al. proposed an adaptively secure FEIP based on DDH and LWE. In [11], Bishop et al. proposed an FEIP supporting function privacy based on Symmetric External Diffie-Hellman assumption (SXDH) in the private key setting (in which ecnryption needs the master secret key which is used to generate the user's private key) by using DPVS. Datta et al. proposed a function-private FEIP which achieves more secure notion of function privacy than [11] based on the same assumption as [11] (instead of increasing the DPVS's dimension) in [12]. Moreover, Abdalla et al. proposed 3-slot multi-input functional encryption for inner products in [2].

2 Preliminaries

2.1 Notations

We follow the notations in [20].

Let A be a set, and then $a \xleftarrow{\mathsf{U}} A$ denotes that a is uniformly selected from A. When B is a random variable or distribution, $b \xleftarrow{\mathsf{R}} B$ denotes that b is randomly selected from B according to its distribution. We define $\mathbb{Z}_p := \{0,1,\ldots,p-1\}$ and $\mathbb{Z}_p^\times := \mathbb{Z}_p\backslash\{0\}$. A vector symbol denotes a vector representation over $\mathbb{Z}_p$, e.g., $\vec{x}$ denotes $(x_1,\ldots,x_n)\in\mathbb{Z}_p^n$. For two vectors $\vec{s}=(s_1,\ldots,s_n)\in\mathbb{Z}_p^n$ and $\vec{t}=(t_1,\ldots,t_n)\in\mathbb{Z}_p^n$, $\vec{s}\cdot\vec{t}$ denotes the inner-product $\sum_{i=1}^n s_i t_i \bmod p$. We let $\vec{0}$ be abused as the zero vector in $\mathbb{Z}_q^n$ for any n.

2.2 General Predicates: Non-Monotone Access Structures with Inner-Product Relations

We follow the definitions in [20]. However, the target vector is $(1,0,\ldots,0)$ as in [26] rather than $(1,1,\ldots,1)$ as in [20].

Definition 1 (Span Programs [20])**.** *Let $\{q_1,\ldots,q_n\}$ be a set of variables. A span program over $\mathbb{Z}_p$ is a labeled matrix $\hat{M}:=(M,\rho)$ where M is a $(\ell\times r)$ matrix over $\mathbb{Z}_p$ and ρ is a labeling of the rows of M by literals from $\{q_1,\ldots,q_n,\neg q_1,\ldots,\neg q_n\}$ (every row is labeled by one literal), i.e., $\rho:\{1,\ldots,\ell\}\to\{q_1,\ldots,q_n,\neg q_1,\ldots,\neg q_n\}$. A span program accepts or rejects an input by the following criterion. For every input sequence $\delta\in\{0,1\}^n$ define the submatrix M_δ of M consisting of those rows whose labels are set to 1 by the input δ, i.e., either rows labeled by some q_i such that $\delta_i=1$ or rows labeled by some $\neg q_i$ such that $\delta_i=0$. (i.e., $\gamma:\{1,\ldots,\ell\}\to\{0,1\}$ is defined by $\gamma(j)=1$ if $[\rho(j)=q_i]\wedge[\delta_i=1]$ or $[\rho(j)=\neg q_i]\wedge[\delta_i=0]$, and $\gamma(j)=0$ otherwise. $M_\delta:=(M_j)_{\gamma(j)=1}$, where M_j is the j-th row of M.)*

We define a target vector $\vec{T}=(1,0,\ldots,0)$. The span program $\hat{M}$ accepts δ if and only if $\vec{T}\in\mathsf{span}\langle M_\delta\rangle$, i.e., some linear combination of the rows of M_δ gives $\vec{T}$. A span program computes a Boolean function f if it accepts exactly those inputs δ where $f(\delta)=1$.

A span program is called monotone if the labels of the rows are only the positive literals $\{q_1,\ldots,q_n\}$. Monotone span programs compute monotone functions. (So, a span program in general is "non"-monotone.)

We assume that no row $M_i (i = 1,\ldots,\ell)$ of the matrix M is $\vec{0}$. We now introduce a non-monotone access structure with evaluating map γ by using the inner-product of attribute vectors, that is employed in our proposed scheme.

Definition 2 (Inner-Products of Attribute Vectors and Access Structures [20]). *$\mathscr{U}_t (t = 1,\ldots,d$ and $\mathscr{U}_t \subset \{0,1\}^*)$ is a sub-universe, a set of attributes, each of which is expressed by a pair of sub-universe id and n_t-dimentional vector, i.e., $(t,\vec{v})$, where $t \in \{1,\ldots,d\}$ and $\vec{v} \in \mathbb{Z}_p^{n_t} \setminus \{\vec{0}\}$.*

We now define such an attribute to be a variable p' of a span program $\hat{M} := (M,\rho)$, i.e., $p' := (t,\vec{v})$ where t is sometimes called category and $\vec{v}$ is called attribute vector. An access structure $\mathbb{S}$ is a span program $\hat{M} := (M,\rho)$ along with variables $q := (t,\vec{v}), q' := (t',\vec{v}'),\ldots$, i.e., $\mathbb{S} := (M,\rho)$ such that $\rho : \{1,\ldots,\ell\} \to \{(t,\vec{v}),(t',\vec{v}'),\ldots,\neg(t,\vec{v}),\neg(t',\vec{v}'),\ldots\}$. Let Γ be a set of attributes, i.e., $\Gamma := \{(t,\vec{x}_t) \mid \vec{x}_t \in \mathbb{Z}_p^{n_t} \setminus \{\vec{0}\}, 1 \le t \le d\}$, where t runs through some subset of $\{1,\ldots,d\}$, not necessarily the whole indices.

When Γ is given to access structure $\mathbb{S}$, map $\gamma : \{1,\ldots,\ell\} \to \{0,1\}$ for span program $\hat{M} := (M,\rho)$ is defined as follows: For $i = 1,\ldots,\ell$, set $\gamma(i) = 1$ if $[\rho(i) = (t,\vec{v}_i)] \wedge [(t,\vec{x}_t) \in \Gamma] \wedge [\vec{v}_i \cdot \vec{x}_t = 0]$ or $[\rho(i) = \neg(t,\vec{v}_i)] \wedge [(t,\vec{x}_t) \in \Gamma] \wedge [\vec{v}_i \cdot \vec{x}_t \neq 0]$. Set $\gamma(i) = 0$ otherwise. Access structure $\mathbb{S} := (M,\rho)$ accepts Γ iff $\vec{T} = (1,0,\ldots,0) \in \mathsf{span}\langle (M_i)_{\gamma(i)=1} \rangle$.

Definition 3 (Linear Secret Sharing Schemes [20]). *A secret sharing scheme for span program $\hat{M} := (M,\rho)$ is:*

1. *Let M be an $\ell \times r$ matrix. Let the target vector $\vec{T} = (1,0,\ldots,0)$ and the column vector $\vec{f}^T := (f_1,\ldots,f_r)^T \xleftarrow{\cup} \mathbb{Z}_q^r$. Then, $s_0 := f_1$ is a secret to be shared, and $\vec{s}^T := (s_1,\ldots,s_\ell)^T := M \cdot \vec{f}^T$ is the vector of ℓ shares of the secret s_0 and the shrae s_i belongs to $\rho(i)$.*

2. *If span program $\hat{M} := (M,\rho)$ accepts δ, or access structure $\mathbb{S} := (M,\rho)$ accepts Γ, i.e., $\vec{T} \in \mathsf{span}\langle (M_i)_{\gamma(i)=1} \rangle$ with $\gamma : \{1,\ldots,\ell\} \to \{0,1\}$, then there exist constants $\{\omega_i \in \mathbb{Z}_p \mid i \in I\}$ such that $I \subseteq \{i \in \{1,\ldots\ell\} \mid \gamma(i) = 1\}$ and $\sum_{i \in I} \omega_i s_i = s_0$. Furthermore, these constants $\{\omega_i\}$ can be computed in time polynomial in the size of matrix M.*

2.3 Symmetric bilinear pairing groups

Definition 4 (Symmetric bilinear pairing groups). *"Symmetric bilinear pairing groups" $(p,\mathbb{G},\mathbb{G}_T,g,e)$ are a tuple of a prime p, cyclic multiplicative group $\mathbb{G}$, $\mathbb{G}_T$ of order p, $g \neq 1 \in \mathbb{G}$, and a polynomial-time computable nondegenerate bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ i.e., $e(g^s,g^t) = e(g,g)^{st}$ and $e(g,g) \neq 1$. Let $\mathscr{G}_{\mathsf{bpg}}$ be an algorithm that takes input 1^λ and outputs a description of bilinear pairing groups $(q,\mathbb{G},\mathbb{G}_T,G,e)$ with security parameter λ.*

2.4 Security Assumption

Definition 5 (q-DBDHE: q-Decisional Bilinear Diffie-Hellman Exponent Assumption)**.** *The q-DBDHE problem is to guess* $\tilde{b} \in \{0,1\}$*, given* $(\mathsf{param}_{\mathbb{G}}, \vec{y}, T_{\tilde{b}}) \xleftarrow{\mathsf{R}} \mathscr{G}_{\tilde{b}}^{\mathsf{q\text{-}DBDHE}}(1^{\lambda})$*, where*

$$
\begin{aligned}
&\mathscr{G}_{\tilde{b}}^{\mathsf{q\text{-}DBDHE}}(1^{\lambda}) : \\
&\mathsf{param}_{\mathbb{G}} := (p, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{\mathsf{R}} \mathscr{G}_{\mathsf{bpg}}(1^{\lambda}), \\
&a, s \xleftarrow{\mathsf{U}} \mathbb{Z}_p,\ \vec{y} := (g^a, \ldots, g^{(a^q)}, g^{(a^{q+2})}, \ldots, g^{(a^{2q})}, g^s), \\
&T_0 := e(g,g)^{(a^{q+1})s}, T_1 := R \xleftarrow{\mathsf{U}} \mathbb{G}_T, \\
&\textit{return } (\mathsf{param}_{\mathbb{G}},\ \vec{y},\ T_{\tilde{b}}),
\end{aligned}
$$

for $\tilde{b} \xleftarrow{\mathsf{U}} \{0,1\}$*. For a probabilistic machine* $\mathscr{B}$*, we define the advantage of* $\mathscr{B}$ *for the q-DBDHE problem as:*
$\mathsf{Adv}_{\mathscr{B}}^{\mathsf{q\text{-}DBDHE}}(\lambda) := |\Pr[\mathscr{B}(1^{\lambda}, \rho) \to 0 \mid \rho \xleftarrow{\mathsf{R}} \mathscr{G}_0^{\mathsf{q\text{-}DBDHE}}(1^{\lambda})] - \Pr[\mathscr{B}(1^{\lambda}, \rho) \to 0 \mid \rho \xleftarrow{\mathsf{R}} \mathscr{G}_1^{\mathsf{q\text{-}DBDHE}}(1^{\lambda})]|$.

The q-DBDHE assumption is: For any probabilistic polynomial-time adversary $\mathscr{B}$*, the advantage* $\mathsf{Adv}_{\mathscr{B}}^{\mathsf{q\text{-}DBDHE}}(\lambda)$ *is negligible in* λ.

2.5 Ciphertext-Policy Attribute-Based Encryption

We describe the syntax of EABEFD, and see Appendix A for the original syntax of general CP-ABE.

2.6 Functional Encryption for Inner Products

Our construction is in part based on the functional encryption for inner products (FEIP) proposed by Abdalla et al. [1] and see Appendix B for its details.

3 Expressive Attribute-Based Encryption with Fast Decryption (EABEFD)

3.1 Definitions of EABEFD

Definition 6 (Expressive Attribute-Based Encryption with Fast Decryption)**.** *An expressive attribute-based encryption with fast decryption (EABEFD) scheme consists of the following algorithms. These are randomized algorithms except for* Dec*.*

1. $\mathsf{Setup}(1^{\lambda}, \vec{n})$

   Setup *algorithm takes as input a security parameter* λ *and format* $\vec{n} := (n_1, \ldots, n_d)$*. It outputs a pair of public parameter and master secret key* (PK, MSK)*.*

2. $\mathsf{KeyGen}(PK, MSK, \Gamma)$

   KeyGen *takes as input a public key PK, master secret key MSK and a set of attributes* $\Gamma := \{(t, \vec{x}_t) \mid \vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t}) \in \mathbb{Z}_p^{n_t} \setminus \{\vec{0}\};\ t \in \{1, \ldots, d\}\ ; x_{t,1} = 1\}$*. It outputs a user private key* sk_{Γ}*.*

3. $\mathsf{Enc}(PK, m, \mathbb{A})$

   Enc *takes as inputs a public key PK, a plaintext m and an access structure* $\mathbb{A} := (M, \rho)$*. It outputs a ciphertext* $CT_{\mathbb{A}}$*.*

4. $\mathsf{Dec}(PK, sk_{\Gamma}, CT_{\mathbb{A}})$

   Dec *takes as inputs a public key PK, a user secret key* sk_{Γ} *and a ciphertext* $CT_{\mathbb{A}}$*. It outputs a message m or a special symbol* $\perp$*.*

An EABEFD scheme should have the following correctness property: for all security parameter λ, all attribute sets $\Gamma := \{(t, \vec{x}_{A,t})\}$, all messages m and all access structures $\mathbb{A}$, it holds that $m = \mathsf{Dec}(PK, sk_\Gamma, CT_\mathbb{A})$ with overwhelming probability, if $\mathbb{A}$ accepts Γ where

$$(PK, MSK) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda, \vec{n} := (n_1, \ldots, n_d)),$$
$$sk_\Gamma \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(PK, MSK, \Gamma),$$
$$CT_\mathbb{A} \xleftarrow{\mathsf{R}} \mathsf{Enc}(PK, m, \mathbb{A})$$

Definition 7 (Restricted-selectively Payload-hiding Secure against the Chosen Plaintext Attack)**.** *For an adversary $\mathscr{A}$, we define $\mathsf{Adv}_{\mathscr{A}}^{\mathsf{EABEFD,r\text{-}PH}}(\lambda) := |\Pr[\mu' = \mu] - 1/2|$ to be the advantage of an adversary in the following experiment for any security parameter λ. An EABEFD scheme is restricted-selectively payload-hiding secure against the chosen plaintext attack if the advantage of any polynomial-time adversary is negligible:*

Init

The adversary $\mathscr{A}$ gives the challenge access structure $\mathbb{A}^* := (M^*, \rho^*)$ to the challenger $\mathscr{C}$. Here, the number of rows of M^* is ℓ^*.

Setup

Given 1^λ, $\mathscr{C}$ runs $\mathsf{Setup}(1^\lambda, \vec{n} := (n_1, \ldots, n_d))$ and gives PK to $\mathscr{A}$.

Phase 1

Let $\tilde{\rho}^* : \{1, \ldots, \ell^*\} \to \{1, \ldots, d\}$ by $\tilde{\rho}(i)^* := t$ if $\rho^*(i) = (t, \vec{v})$ or $\rho^*(i) = \neg(t, \vec{v})$, where ρ^* is given in challenge access structure $\mathbb{A}^*$. $\mathscr{A}$ is allowed to issue a polynomial number of key queries for some attribute sets $\Gamma_i := \{(t, \vec{x}_t)\} (i = 1, \ldots, q_1)$ to $\mathscr{C}$. Then, $\mathscr{C}$ runs $\mathsf{KeyGen}(PK, MSK, \Gamma_i)$ for $i = 1, \ldots, q_1$ to generate attribute secret key sk_{Γ_i} and gives it to $\mathscr{A}$. Here, for any Γ_i where $i = 1, \ldots, q_1$, the following must hold: $\vec{T} \notin \mathsf{span}\langle (M^*_{i'})_{i' \in I'} \rangle$ where $M^*_{i'}$ is the i'-th row of M^* and I' is defined as $\{i' \mid 1 \leq i' \leq \ell^* \text{ and } \exists (t, \vec{x}_t) \in \Gamma_i, \tilde{\rho}^*(i') = t\}$.

Challenge

$\mathscr{A}$ gives two challenge plaintexts m_0^*, m_1^* to $\mathscr{C}$. $\mathscr{C}$ flips a random coin $\mu \xleftarrow{\mathsf{U}} \{0, 1\}$, and computes $CT_{\mathbb{A}^*} \xleftarrow{R} \mathsf{Enc}(PK, m_\mu^*, \mathbb{A}^*)$. Then, $\mathscr{C}$ gives $CT_{\mathbb{A}^*}$ to $\mathscr{A}$.

Phase 2

$\mathscr{A}$ is allowed to issue a polynomial number of key queries for some attribute sets $\Gamma_i (i = q_1 + 1, \ldots, \nu)$ to $\mathscr{C}$ as in **Phase1**. *Then, $\mathscr{C}$ runs $\mathsf{KeyGen}(PK, MSK, \Gamma_i)$ for $i = q_1 + 1, \ldots, \nu$ to generate attribute secret key sk_{Γ_i} and gives it to $\mathscr{A}$. Here, for any Γ_i where $i = 1, \ldots, q_1, q_1 + 1, \ldots, \nu$, the following must hold: $\vec{T} \notin \mathsf{span}\langle (M^*_{i'})_{i' \in I'} \rangle$ where $M^*_{i'}$ is the i'-th row of M^* and I' is defined as $\{i' \mid 1 \leq i' \leq \ell^* \text{ and } \exists (t, \vec{x}_t) \in \Gamma_i, \tilde{\rho}^*(i') = t\}$.*

Guess

$\mathscr{A}$ outputs a guess μ' of μ. If $\mu' = \mu$, then $\mathscr{A}$ wins.

Remark: We compare Definitions 9 (selectively payload-hiding security) and 7 (restricted-selectively payload-hiding security). Definition 7 is the restricted Definition 9 in that it has the stronger restriction, i.e., $\vec{T} \notin \mathsf{span}\langle (M^*_j)_{j \in I'} \rangle$ for any Γ_i in **Phase 1** and **Phase 2**. Intutively, this stronger restriction limits the key query in terms of the category.

For example, we assume that there are attribute categories t_1 and t_2 in the ABE system. t_1 includes attributes a_1 and b_1 (or attribute vectors $(t_1, \vec{a}_1)$ and $(t_1, \vec{b}_1)$ respcetively). t_2 includes attributes a_2 and

b_2 (or attribute vectors $(t_2,\vec{a}_2)$ and $(t_2,\vec{b}_2)$ respectively). We also assume the challenge access policy is $a_1 \wedge a_2$. In the selectively payload-hiding security game, the adversary can query the private keys for $\{a_1,b_2\}$, $\{b_1,b_2\}$, $\{b_1,a_2\}$ or etc., but cannot query the key for $\{a_1,a_2\}$ because of key query restriction ($\mathbb{A}^*$ does not accept Γ_i). In the restricted-selectively payload-hiding security game, the adversary can query the private keys for $\{(t_1,\vec{a}_1)\}$, $\{(t_1,\vec{b}_1)\}$, $\{(t_2,\vec{a}_2)\}$ or $\{(t_2,\vec{b}_2)\}$. However, he/she cannot query the keys for $\{(t_1,\vec{a}_1),(t_2,\vec{a}_2)\}$, $\{(t_1,\vec{a}_1),(t_2,\vec{b}_2)\}$, $\{(t_1,\vec{b}_1),(t_2,\vec{b}_2)\}$ and $\{(t_1,\vec{b}_1),(t_2,\vec{a}_2)\}$ because of the restriction ($\vec{T} \notin \mathsf{span}\langle(M^*_j)_{j\in I'}\rangle$). In this case, I' is $\{j \mid \tilde{\rho}(j) = t_1\}$ or $\{j \mid \tilde{\rho}(j) = t_2\}$. Therefore, the adversary can query the private keys for $\{(t_1,\vec{a}_1)\}$, $\{(t_1,\vec{b}_1)\}$, $\{(t_2,\vec{a}_2)\}$ or $\{(t_2,\vec{b}_2)\}$ only.

Here, we note that the security of our scheme supporting non-monotone access structure with inner-product relations is *not* weaker than that of the previous scheme supporting monotone access structure [26]. That is, our scheme supporting *only* monotone access structure *without* inner-product relation achieves not only the restricted-selectively payload-hiding security (Definition 7) but also the selectively payload-hiding security (Definition 9). In the example as we have shown, if we assume (t_1,a_1), (t_2,a_2), (t_3,b_1) and (t_4,b_2), both the adversary in Definition 9 and the one in Definition 7 can query the private keys for $\{a_1,b_2\}$, $\{b_1,b_2\}$, $\{b_1,a_2\}$ or etc., but cannot query the key for $\{a_1,a_2\}$ where the chllenge access policy is $a_1 \wedge a_2$. It is because $\tilde{\rho}$ in our scheme supporting only access structure without inner-product relation equals to ρ in the previous scheme [26]. Therefore, our scheme achieves at least the same level offered by the previous scheme [26]. That is, our scheme supporting non-monotone access structures with inner product relations achieves the intermediate security between the one of the previous scheme [26] and the one in Definition 9 where the access structure is non-monotone with inner-product relations.

3.2 Construction

Let be $\tilde{\rho} : \{1,\ldots,\ell\} \to \{1,\ldots,d\}$ by $\tilde{\rho}(i) := t$ if $\rho(i) = (t,\vec{v})$ or $\rho(i) = \neg(t,\vec{v})$, where ρ is given in access structure $\mathbb{A} := (M,\rho)$. We note that our proposal works with the restriction that an attribute can only be used in at most one row in the access matrix M as well as the scheme [26]. That is, the $\tilde{\rho}$ function is injective. As in [20], we assume that input vector $\vec{x}_t := (x_{t,1},\ldots,x_{t,n_t})$ is normalized such that $x_{t,1} := 1$. We also assume that $x_{t,1}$ is non-zero. If $\vec{x}_t$ is not normalized, we can change it to a normalized one by $(1/x_{t,1})\cdot\vec{x}_t$.

1. $\mathsf{Setup}(1^\lambda,\vec{n} := (d,n_1,\ldots,n_d))$:

   $\mathsf{param}_{\mathbb{G}} := (p,\mathbb{G},\mathbb{G}_T,g,e) \xleftarrow{\mathsf{R}} \mathscr{G}_{\mathsf{bpg}}(1^\lambda)$, $\alpha,\beta,a \xleftarrow{\mathsf{U}} \mathbb{Z}_p^\times$, $\mathsf{gparam} := (\mathsf{param}_{\mathbb{G}}, e(g,g)^\alpha, g^a, g^\beta)$,

   $h_t \xleftarrow{\mathsf{U}} \mathbb{G}$ for $t = 1,\ldots,d$,

   $s_{t,j} \xleftarrow{\mathsf{U}} \mathbb{Z}_p^\times$ for $t = 1,\ldots,d; j = 1,\ldots,n_t$,

   $PK := (\mathsf{gparam},\ \{h_t\}_{t=1}^d,\ \{g^{s_{t,j}}\}_{t=1;j=1}^{d;n_t})$, $MSK := (\{\vec{s}_t := (s_{t,1},\ldots,s_{t,n_t})\}_{t=1}^d, g^\alpha, \beta)$,

   return (PK,MSK).

2. $\mathsf{KeyGen}(PK,MSK,\Gamma = \{(t,\vec{x}_t) \mid \vec{x}_t := (x_{t,1},\ldots,x_{t,n_t}) \in \mathbb{Z}_p^{n_t} \setminus \{\vec{0}\}; 1 \le t \le d; x_{t,1} = 1\})$:

   $z \xleftarrow{\mathsf{U}} \mathbb{Z}_p^\times$, $K := g^\alpha g^{az}$, $L := g^z$, $K_t := h_t^z$ for t s.t. $(t,\vec{x}_t) \in \Gamma$,

   for $(t,\vec{x}_t) \in \Gamma$,

   $\tau_{(t,\vec{x}_t)} \xleftarrow{\mathsf{U}} \mathbb{Z}_p^\times$,

   $k_{(t,\vec{x}_t)} := \tau_{(t,\vec{x}_t)}(\sum_{j=1}^{n_t} x_{t,j}s_{t,j}) = \tau_{(t,\vec{x}_t)}\vec{x}_t \cdot \vec{s}_t$,

$\tilde{k}_{(t,\vec{x}_t)} := g^{z(1-\tau_{(t,\vec{x}_t)})(\sum_{j=1}^{n_t} x_{t,j}s_{t,j})/\beta} = g^{z(1-\tau_{(t,\vec{x}_t)})\vec{x}_t\cdot\vec{s}_t/\beta}$,

return $sk_\Gamma := (\Gamma, K, L, \{K_t\}_{t \text{ s.t. } (t,\vec{x}_t)\in\Gamma}, \{k_{(t,\vec{x}_t)}, \tilde{k}_{(t,\vec{x}_t)}\}_{(t,\vec{x}_t)\in\Gamma})$.

Remark: We explain how to modify FEIP [1] briefly. We use two parameters $\tau_{(t,\vec{x}_t)}$ and β. $\tau_{(t,\vec{x}_t)}$ is the value to prevent users from colluding and getting $\vec{s}_t$ by using $k_{(t,\vec{x}_t)}$ and solving simultuneous equations. β is the value to connect CP-ABE [16] and FEIP [1].

3. $\mathsf{Enc}(PK, m, \mathbb{A} = (M, \rho))$:

$\vec{f} := (s, y_2, \ldots, y_r)$, $\vec{\lambda} := (\lambda_1, \ldots, \lambda_\ell)^T = M\cdot\vec{f}^T$, $r' \xleftarrow{\cup} \mathbb{Z}_p^\times$,

$C := me(g,g)^{\alpha s}$, $C' := g^s$, $c := g^{r'}$, $\tilde{c} := g^{\beta r'}$,

for $i = 1, \ldots, \ell$,

$\theta_i \xleftarrow{\cup} \mathbb{Z}_p$,

if $\rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{Z}_p^{n_t} \setminus \{\vec{0}\})$,

$c_{i,1} := g^{r's_{t,1}}(g^{a\lambda_i}h_t^{-s})g^{\theta_i v_{i,1}}$, $c_{i,2} := g^{r's_{t,2}}g^{\theta_i v_{i,2}}$, $\ldots$ $c_{i,n_t} := g^{r's_{t,n_t}}g^{\theta_i v_{i,n_t}}$,

if $\rho(i) = \neg(t, \vec{v}_i)$,

$c_{i,1} := g^{r's_{t,1}}(g^{a\lambda_i}h_t^{-s})^{v_{i,1}}$, $c_{i,2} := g^{r's_{t,2}}(g^{a\lambda_i}h_t^{-s})^{v_{i,2}}$, $\ldots$ $c_{i,n_t} := g^{r's_{t,n_t}}(g^{a\lambda_i}h_t^{-s})^{v_{i,n_t}}$,

return $CT_\mathbb{A} := (\mathbb{A}, C, C', c, \tilde{c}, \{c_{i,j}\}_{i=1;j=1}^{\ell;n_{\tilde{\rho}(i)}})$.

4. $\mathsf{Dec}(PK, sk_\Gamma, CT_\mathbb{A})$:

If $\mathbb{A} := (M, \rho)$ accepts $\Gamma := \{(t, \vec{x}_t)\}$, then compute I and $\{\omega_i\}_{i\in I}$ s.t. $\vec{T} = \sum_{i\in I}\omega_i M_i$, where M_i is the i-th row of M, and

$I \subseteq \{i \in \{1,\ldots,\ell\} \mid [\rho(i) = (t,\vec{v}_i) \wedge (t,\vec{x}_t)\in\Gamma \wedge \vec{v}_i\cdot\vec{x}_t = 0] \vee [\rho(i) = \neg(t,\vec{v}_i) \wedge (t,\vec{x}_t)\in\Gamma \wedge \vec{v}_i\cdot\vec{x}_t \neq 0]\}$,

for $i \in I$,

$K'_i := (\prod_{j=1}^{n_t}(c_{i,j})^{x_{t,j}})/c^{k_{(t,\vec{x}_t)}}$,

$$K' := e\Big(\prod_{i\in I\wedge\rho(i)=(t,\vec{v}_i)} {K'_i}^{-\omega_i} \prod_{i\in I\wedge\rho(i)=\neg(t,\vec{v}_i)} {K'_i}^{-\omega_i/(\vec{v}_i\cdot\vec{x}_t)},\ L\Big) \cdot e\Big(\tilde{c}, \prod_{i\in I\wedge\rho(i)=(t,\vec{v}_i)} \tilde{k}_{(t,\vec{x}_t)}^{\omega_i} \prod_{i\in I\wedge\rho(i)=\neg(t,\vec{v}_i)} \tilde{k}_{(t,\vec{x}_t)}^{\omega_i/(\vec{v}_i\cdot\vec{x}_t)}\Big)\cdot e\Big(C',\ K\prod_{i\in I}K_{\tilde{\rho}(i)}{}^{\omega_i}\Big)$$

return C/K'.

Correctness :

$$K'_i = \begin{cases} g^{r'(1-\tau_{(t,\vec{x}_t)})\vec{x}_t\cdot\vec{s}_t}(g^{a\lambda_i}h_i^{-s}) & (i\in I; \rho(i) = (t,\vec{v}_i)) \\ g^{r'(1-\tau_{(t,\vec{x}_t)})\vec{x}_t\cdot\vec{s}_t}(g^{a\lambda_i}h_i^{-s})^{\vec{v}_i\cdot\vec{x}_t} & (i\in I; \rho(i) = \neg(t,\vec{v}_i)) \end{cases},$$

$$K' = \prod_{i\in I} e(g,g)^{-az\omega_i\lambda_i} e(g, h_{\tilde{\rho}(i)})^{-sz\omega_i} \cdot \prod_{i\in I\wedge\rho(i)=(t,\vec{v}_i)} e(g,g)^{-\omega_i r'z(1-\tau_{(t,\vec{x}_t)})\vec{x}_t\cdot\vec{s}_t} \prod_{i\in I\wedge\rho(i)=\neg(t,\vec{v}_i)} e(g,g)^{-(\omega_i/\vec{v}_i\cdot\vec{x}_t)r'z(1-\tau_{(t,\vec{x}_t)})\vec{x}_t\cdot\vec{s}_t} \cdot \prod_{i\in I\wedge\rho(i)=(t,\vec{v}_i)} e(g,g)^{\omega_i r'z(1-\tau_{(t,\vec{x}_t)})\vec{x}_t\cdot\vec{s}_t} \prod_{i\in I\wedge\rho(i)=\neg(t,\vec{v}_i)} e(g,g)^{(\omega_i/\vec{v}_i\cdot\vec{x}_t)r'z(1-\tau_{(t,\vec{x}_t)})\vec{x}_t\cdot\vec{s}_t}$$

$$\cdot e(g,g)^{s\alpha} e(g,g)^{saz} \prod_{i\in I} e(g, h_{\tilde{\rho}(i)})^{sz\omega_i}$$
$$= e(g,g)^{\alpha s} \quad \text{since} \sum_{i\in I} \omega_i \lambda_i = s.$$

The security proof of our construction can be found in Appendix C. In Appendix C, we explain why our construction can achieve only r-PH security against the chosen plaintext attacks instead of selectively payload-hiding security against the chosen plaintext attacks in [26] when an attribute category has some attributes and/or an access structure is non-monotone.

4 Performance

Table 1: Comparison of Performance with Previous Works: $|I|$: the number of attributes which are used for decryption, $\hat{n}$: the maximum number of dimension of attribute vectors in private keys which are used when a user decrypts a ciphertext

Schemes	Computation Cost for Decryption		Access structure
	exp.	pairing	
HW13 [16, Section 3.5] (W11 [26, Section 5])	$O(\|I\|)$	2	Monotone(LSSS)
OT10 [20]	$O(\|I\|)$	$O(\|I\|\hat{n})$	Non-monotone (LSSS & Inner-Product)
ZZCLL14 [27]	-	2	AND(multiple values & wildcard)
MST17 [18] (LSSS)	$O(\|I\|)$	2	Monotone (LSSS)
MST17 [18] (DNF)	-	2	Monotone (DNF)
AC17[4]	-	6	Monotone (LSSS)
This work	$O(\|I\|\hat{n})$	3	Non-monotone (LSSS & Inner-Product)

Table 2: Comparison of Performance with Previous Works: LSSS: linear secret sharing scheme, DNF: disjunctive normal form, STD: standard model, ROM: random oracle model

Schemes	Security Model	Assumption
HW13 [16, Section 3.5] (W11 [26, Section 5])	selective (STD)	q-DBDHE
OT10 [20]	adaptive (STD)	DLIN
ZZCLL14 [27]	selective (ROM)	q-DBDHE
MST17 [18] (LSSS)	selective (STD)	Modified-BDHE
MST17 [18] (DNF)	selective (STD)	Modified-BDHE
AC17[4]	adaptive (ROM)	DLIN
This work	r-selective (STD)	q-DBDHE

We show a comparison of performance with previous works in Tables 1, 2, 3 and 4.

In Tables 1 and 2, "exp." and "pairing" mean the number of exponentiations and pairing operations on elements of $\mathbb{G}$. The scheme [20] is built in the additive group, so "exp." means the number of constant multiple calculations in OT10 [20]. $|I|$ represents the number of attributes which are used for decryption. $\hat{n}$ represents the maximum number of dimension of attribute vectors in private keys which are used when a user decrypts a ciphertext. LSSS and DNF mean linear secret sharing scheme and disjunctive normal

Table 3: Comparison of Performance with Previous Works (cont.): $|\mathbb{G}|, |\mathbb{G}_T|$ and $|\mathbb{Z}_p^\times|$: the bit length of the element of $\mathbb{G}, \mathbb{G}_T$ and $\mathbb{Z}_p^\times$ respectively, $|\Gamma|$: the number of attributes in private keys, n: the maximum number of dimension of attribute vectors in private keys, d: the number of attributes which are managed by the KGC, n_{max}: the maximum number of dimension of attribute vectors which are managed by the KGC, ℓ: the number of rows of the access matrix, $\tilde{n}$: the maximum number of dimension of attribute vectors in ciphertexts, m: the number of clauses in a DNF

Schemes	Parameter Size			
	SK	PK	MSK	CT
HW13 [16, Section 3.5] (W11 [26, Section 5])	$O(\lvert\Gamma\rvert)\lvert\mathbb{G}\rvert$	$O(d)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$\lvert\mathbb{G}\rvert$	$O(\ell)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$
OT10 [20]	$O(\lvert\Gamma\rvert n)\lvert\mathbb{Z}_p\rvert$+$O(\lvert\Gamma\rvert n)\lvert\mathbb{G}\rvert$	$O(dn_{max})\lvert\mathbb{G}\rvert$	$O(dn_{max})\lvert\mathbb{G}\rvert$	$O(\ell\tilde{n})\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$
ZZCLL14 [27]	$O(\lvert\Gamma\rvert)\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_p^\times\rvert$	$O(d)\lvert\mathbb{G}\rvert + O(d)\lvert\mathbb{G}_T\rvert$	$2\lvert\mathbb{Z}_p^\times\rvert$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$
MST17 [18] (using LSSS)	$O(\lvert\Gamma\rvert)\lvert\mathbb{G}\rvert$	$O(d)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$\lvert\mathbb{G}\rvert$	$O(\ell)\lvert\mathbb{G}\rvert$
MST17 [18] (using DNF)	$O(\lvert\Gamma\rvert)\lvert\mathbb{G}\rvert$	$O(d)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$\lvert\mathbb{G}\rvert$	$O(m)\lvert\mathbb{G}\rvert$
AC17 [4]	$O(\lvert\Gamma\rvert)\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{G}\rvert + 2\lvert\mathbb{G}_T\rvert$	$4\lvert\mathbb{G}\rvert + 4\lvert\mathbb{Z}_p^\times\rvert$	$O(\ell)\lvert\mathbb{G}\rvert$
This work	$O(\lvert\Gamma\rvert n)\lvert\mathbb{Z}_p\rvert$+$O(\lvert\Gamma\rvert)\lvert\mathbb{G}\rvert$	$O(dn_{max})\lvert\mathbb{G}\rvert$	$O(dn_{max})\lvert\mathbb{Z}_p^\times\rvert$	$O(\ell\tilde{n})\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$

Table 4: Comparison of Performance with Previous Works (cont.): $\tilde{n}$, ℓ and m: the seme as in Table 3, r: the number of columns of the access matrix

Schemes	Computation Cost for Encryption	
	hash	exp.
HW13 [16, Section 3.5] (W11 [26, Section 5])	-	$O(\ell)$
OT10 [20]	-	$O(\ell\tilde{n}^2)$
ZZCLL14 [27]	-	2
MST17 [18] (using LSSS)	-	$O(\ell)$
MST17 [18] (using DNF)	-	$O(m)$
AC17 [4]	$O(\ell r)$	$O(\ell r)$
This work	-	$O(\ell\tilde{n})$

form. In the security model column, selective/adaptive/r-selective mean selectively/adaptively/restricted-selectively payload-hiding security against the chosen plaintext attack respectively. STD and ROM mean standard model and random oracle model.

In Table 3, SK, PK, MSK and CT represent the bit length of private key, public key, master secret key and ciphertext, respectively. $|\mathbb{G}|, |\mathbb{G}_T|$ and $|\mathbb{Z}_p^\times|$ represent the bit length of the element of $\mathbb{G}, \mathbb{G}_T$ and $\mathbb{Z}_p^\times$, respectively. $|\Gamma|$ is the number of attributes in private keys. n represents the maximum number of dimension of attribute vectors in private keys. d represents the number of attributes which are managed by the KGC. n_{max} represents the maximum number of dimension of attribute vectors which are managed by the KGC. ℓ is the number of rows of the access matrix. $\tilde{n}$ replesents the maximum number of dimension of attribute vectors in ciphertexts. m replesents the number of clauses in a DNF.

In Table 4, "exp." means the same as in Table 1. "hash" means the number of hash operations which map arbitrary binary strings to elements of $\mathbb{G}$. $\tilde{n}$, ℓ and m replesent the seme as in Table 3. r is the number of columns of the access matrix.

Tables 1 and 2 shows that our proposal is a trade-off variant of OT10 [20] (which can be viewed as one of the most expressive CP-ABE schemes) because the cost of pairing operations is more expensive than that of exponentiations on elements of $\mathbb{G}$ [15]. Table 1 also shows that no schemes achieve a constant

number of pairing operations and non-monotone access structure simultaneously except our proposal. Table 3 shows that the proposal is more efficient in terms of the size of SK, MSK and CT than [20]. Table 4 shows that our proposal is more efficient in terms of the number of exp. than [20].

5 Conclusion

In this paper, we proposed the new CP-ABE scheme supporting non-monotone access structures with inner-product relations, which needs only three pairing operations for decryption. There is no CP-ABE scheme that supports non-monotone access structures with constant pairing operations for decryption except our proposal as far as we know. We also introduced a new security model, r-PH security against the chosen plaintext attacks that is weaker than selectively payload-hiding security against the chosen plaintext attacks in [26]. We proved that our construction achieves r-PH security under the q-DBDHE assumption.

Our construction can be viewed as the trade-off variant of [20]. Even if some users own the only devices which have only low computation resources, they can be expected to use the CP-ABE in our proposal. Extending our scheme such that $\tilde{\rho}$ does not need to be injective and proving that our scheme achieves selective security in the model stronger than restricted-selective security are left as future work.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Number 17K00178 and the Telecommunications Advancement Foundation.

References

[1] M. Abdalla, F. Bourse, A. D. Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In *Proc. of the 18th IACR International Conference on Practice and Theory in Public-Key Cryptography (PKC'15), Gaithersburg, MD, USA*, volume 9020 of *Lecture Notes in Computer Science*, pages 733–751. Springer, Berlin, Heidelberg, March-April 2015.

[2] M. Abdalla, R. Gay, M. Raykova, and H. Wee. Multi-input inner-product functional encryption from pairings. In *Proc. of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'17), Paris, France*, volume 10210 of *Lecture Notes in Computer Science*, pages 601–626. Springer, Cham, April-May 2017.

[3] S. Agrawal and M. Chase. A study of pair encodings: Predicate encryption in prime order groups. In *Proc. of the 13th International Conference on Theory of Cryptography (TCC'16), Tel Aviv, Israel*, volume 9563 of *Lecture Notes in Computer Science*, pages 259–288. Springer, Berlin, Heidelberg, January 2016.

[4] S. Agrawal and M. Chase. FAME: fast attribute-based message encryption. In *Proc. of the 2017 ACM Conference on Computer and Communications Security (ACM CCS'17), Dallas, Texas, USA*, pages 665–682. ACM, October-November 2017.

[5] S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *Proc. of the 36th Annual International Cryptology Conference (CRYPTO'16), Santa Barbara, CA, USA*, volume 9816 of *Lecture Notes in Computer Science*, pages 333–362. Springer, Berlin, Heidelberg, August 2016.

[6] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R. Moosavi, A. Rahmani, and P. Liljeberg. On the feasibility of attribute-based encryption on internet of things devices. *IEEE Micro*, 36(6):25–35, January-February 2016.

[7] M. Ambrosin, M. Conti, and T. Dargahi. On the feasibility of attribute-based encryption on smartphone devices. In *Proc. of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems (IoT-Sys'15), Florence, Italy*, pages 49–54. ACM, May 2015.

[8] N. Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In *Proc. of the 22nd International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'16), Hanoi, Vietnam*, volume 10032 of *Lecture Notes in Computer Science*, pages 591–623. Springer, Berlin, Heidelberg, December 2016.

[9] N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *Proc. of the 14th International Conference on Practice and Theory in Public Key Cryptography (PKC'11), Taormina, Italy*, volume 6571 of *Lecture Notes in Computer Science*, pages 90–108. Springer, Berlin, Heidelberg, March 2011.

[10] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Proc. of the 2017 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA*, pages 321–334. IEEE, May 2007.

[11] A. Bishop, A. Jain, and L. Kowalczyk. Function-hiding inner product encryption. In *Proc. of the 21st International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'15), Auckland, New Zealand*, volume 9452 of *Lecture Notes in Computer Science*, pages 470–491. Springer, Berlin, Heidelberg, November-December 2015.

[12] P. Datta, R. Dutta, and S. Mukhopadhyay. Functional encryption for inner product with full function privacy. In *Proc. of the 19th IACR International Conference on Practice and Theory in Public-Key Cryptography (PKC'16), Taipei, Taiwan*, volume 9614 of *Lecture Notes in Computer Science*, pages 164–195. Springer, Berlin, Heidelberg, March 2016.

[13] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of the 2006 ACM Conference on Computer and Communications Security (ACM CCS'06), Alexandria, Virginia, USA*, pages 89–98. ACM, October-November 2006.

[14] M. Green, S. Hohenberger, and B. Waters. Outsourcing the decryption of ABE ciphertexts. In *Proc. of the 20th USENIX Conference on Security (SEC'11), San Francisco, CA, USA*. USENIX Association, August 2011.

[15] A. Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In *Proc. of the 11th International Conference on Applied Cryptography and Network Security (ACNS'13), Banff, AB, Canada*, volume 7954 of *Lecture Notes in Computer Science*, pages 357–372. Springer, Berlin, Heidelberg, June 2013.

[16] S. Hohenberger and B. Waters. Attribute-based encryption with fast decryption. In *Proc. of the 16th International Conference on Practice and Theory in Public-Key Cryptography (PKC'13), Nara, Japan*, volume 7778 of *Lecture Notes in Computer Science*, pages 162–179. Springer, Berlin, Heidelberg, February-March 2013.

[17] J. Kim, W. Susilo, F. Guo, M. H. Au, and S. Nepal. An efficient KP-ABE with short ciphertexts in prime ordergroups under standard assumption. In *Proc. of the 2017 ACM on Asia Conference on Computer and Communications Security (AsiaCCS'17), Abu Dhabi, United Arab Emirates*, pages 823–834. ACM, April 2017.

[18] Q. M. Malluhi, A. Shikfa, and V. C. Trinh. A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption. In *Proc. of the 2017 ACM on Asia Conference on Computer and Communications Security (AsiaCCS'17), Abu Dhabi, United Arab Emirates*, pages 230–240. ACM, April 2017.

[19] S. Moffat, M. Hammoudeh, and R. Hegarty. A survey on ciphertext-policy attribute-based encryption (cp-abe) approaches to data security on mobile devices and its application to iot. In *Proc. of the 2017 International Conference on Future Networks and Distributed Systems (ICFNDS'17), Cambridge, UK*, pages 34:1–34:10. ACM, July 2017.

[20] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *Proc. of the 30th Annual Cryptology Conference (CRYPTO'10), Santa Barbara, CA, USA*, volume 6223 of *Lecture Notes in Computer Science*, pages 191–208. Springer, Berlin, Heidelberg, August 2010.

[21] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *Proc. of the 2007 ACM Conference on Computer and Communications Security (ACM CCS'07), Alexandria, VA, USA*, pages 195–203. ACM, October-November 2007.

[22] Y. S. Rao and R. Dutta. Decentralized ciphertext-policy attribute-based encryption scheme with fast decryption. In *Proc. of the 14th IFIP TC 6/TC 11 International Conference (CMS'13), Magdeburg, Germany*, volume 8099 of *Lecture Notes in Computer Science*, pages 66–81. Springer, Berlin, Heidelberg, September 2013.

[23] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Proc. of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'05), Aarhus, Denmark*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, Berlin, Heidelberg, May 2005.

[24] K. Takashima. Expressive attribute-based encryption with constant-size ciphertexts from the decisional linear assumption. In *Proc. of the 9th International Conference (SCN'14), Amalfi, Italy*, volume 8642 of *Lecture Notes in Computer Science*, pages 298–317. Springer, Cham, September 2014.

[25] L. Touati and Y. Challal. Efficient CP-ABE attribute/key management for iot applications. In *Proc. of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT'15/IUCC'15/DASC'15/PICom'15), Liverpool, UK*, pages 343–350. IEEE, October 2015.

[26] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proc. of the 14th International Conference on Practice and Theory in Public Key Cryptography (PKC'11), Taormina, Italy*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer, Berlin, Heidelberg, March 2011.

[27] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li. Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts. In *Proc. of the 8th International Conference on Provable Security (ProvSec'14), Hong Kong, China*, volume 8782 of *Lecture Notes in Computer Science*, pages 259–273. Springer, Cham, October 2014.

Appendix

A CP-ABE

Here, we explain the ciphertext-policy attribute-based encryption (CP-ABE) proposed by Waters [26] briefly.

A.1 Definition of CP-ABE

Definition 8 (Ciphertext-Policy Attribute-Based Encryption)**.** *A ciphertext-policy attribute-based encryption (CP-ABE) scheme consists of the following algorithms. These are randomized algorithms except for* Dec*.*

1. $\mathsf{Setup}(1^\lambda, U)$
   Setup *algorithm takes as input a security parameter* λ *and the attribute universe* U*. It outputs a pair of public parameter and master secret key* (PK, MSK)*.*

2. $\mathsf{KeyGen}(PK, MSK, \Gamma)$
   KeyGen *takes as input a public key PK, master secret key MSK and a set of attributes* Γ*. It outputs a user private key* sk_Γ*.*

3. $\mathsf{Enc}(PK, m, \mathbb{A})$
   Enc *takes as inputs a public key PK, a plaintext m and an access structure* $\mathbb{A} := (M, \rho)$*. It outputs a ciphertext* $CT_{\mathbb{A}}$*.*

4. $\mathsf{Dec}(PK, sk_\Gamma, CT_\mathbb{A})$

   Dec *takes as inputs a public key PK, a user secret key* sk_Γ *and a ciphertext* $CT_\mathbb{A}$*. It outputs a message m or a special symbol* $\perp$*.*

A CP-ABE scheme should have the following correctness property: for all security parameter λ, all attribute sets Γ, all messages m and all access structures $\mathbb{A}$, it holds that $m = \mathsf{Dec}(PK, sk_\Gamma, CT_\mathbb{A})$ with overwhelming probability, if $\mathbb{A}$ accepts Γ where

$$(PK, MSK) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^\lambda, U),$$
$$sk_\Gamma \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(PK, MSK, \Gamma),$$
$$CT_\mathbb{A} \xleftarrow{\mathsf{R}} \mathsf{Enc}(PK, m, \mathbb{A})$$

A.2 Security of CP-ABE

Definition 9 (Selectively Payload-hiding Security against the Chosen Plaintext Attack)**.** *For an adversary* $\mathscr{A}$*, we define* $\mathsf{Adv}_{\mathscr{A}}^{\mathsf{CP\text{-}ABE,PH}}(\lambda) := |\Pr[\mu' = \mu] - 1/2|$ *to be the advantage of an adversary in the following experiment for any security parameter* λ*. A CP-ABE scheme is selectively payload-hiding secure against the chosen plaintext attack if the advantage of any polynomial-time adversary is negligible:*

Init

The adversary $\mathscr{A}$ *gives the challenge access structure* $\mathbb{A}^* := (M^*, \rho^*)$ *to the challenger* $\mathscr{C}$*.*

Setup

Given 1^λ*,* $\mathscr{C}$ *runs* $\mathsf{Setup}(1^\lambda, U)$ *and gives PK to* $\mathscr{A}$*.*

Phase 1

$\mathscr{A}$ *is allowed to issue a polynomial number of key queries for some attribute sets* $\Gamma_i (i = 1, \ldots, q_1)$ *to* $\mathscr{C}$*. Then,* $\mathscr{C}$ *runs* $\mathsf{KeyGen}(PK, MSK, \Gamma_i)$ *for* $i = 1, \ldots, q_1$ *to generate attribute secret key* sk_{Γ_i} *and gives it to* $\mathscr{A}$*. Here, we remark that* $\mathbb{A}^*$ *does not accept* Γ_i *for* $i = 1, \ldots, q_1$*.*

Challenge

$\mathscr{A}$ *gives two challenge plaintexts* m_0^*, m_1^* *to* $\mathscr{C}$*.* $\mathscr{C}$ *flips a random coin* $\mu \xleftarrow{\mathsf{U}} \{0,1\}$*, and computes* $CT_{\mathbb{A}^*} \xleftarrow{R} \mathsf{Enc}(PK, m_\mu^*, \mathbb{A}^*)$*. Then,* $\mathscr{C}$ *gives* $CT_{\mathbb{A}^*}$ *to* $\mathscr{A}$*.*

Phase 2

$\mathscr{A}$ *is allowed to issue a polynomial number of key queries for some attribute sets* $\Gamma_i (i = q_1 + 1, \ldots, \nu)$ *to* $\mathscr{C}$ *as in* **Phase1***. Then,* $\mathscr{C}$ *runs* $\mathsf{KeyGen}(PK, MSK, \Gamma_i)$ *for* $i = q_1 + 1, \ldots, \nu$ *to generate attribute secret key* sk_{Γ_i} *and gives it to* $\mathscr{A}$*. Here, we remark that* $\mathbb{A}^*$ *does not accept* Γ_i *for* $i = 1, \ldots, q_1, \ldots, \nu$*.*

Guess

$\mathscr{A}$ *outputs a guess* μ' *of* μ*. If* $\mu' = \mu$*, then* $\mathscr{A}$ *wins.*

In adaptively payload-hiding security against the chosen plaintext attack [20], there is no **Init** phase and an adeversary issues a challenge access structure in **Challenge** phase, i.e., after key query.

A.3 Construction of CP-ABE

Here we give a construction of [26, Section 5] ([16, Section 3.5]) which our scheme is based on.

1. $\mathsf{Setup}(1^\lambda,\ U)$:

   $\mathsf{param}_{\mathbb{G}} := (p,\mathbb{G},\mathbb{G}_T,g,e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda),\ \alpha,a \xleftarrow{\mathsf{U}} \mathbb{Z}_p,\ h_1,\ldots,h_U \xleftarrow{\mathsf{U}} \mathbb{G}$

   $\mathsf{gparam} := (\mathsf{param}_{\mathbb{G}}, e(g,g)^\alpha, g^a),$

   $PK := (\mathsf{gparam},\ \{h_{i'}\}_{i'=1}^{U}),\ MSK := g^\alpha,$

   return (PK, MSK).

2. $\mathsf{KeyGen}(PK,\ MSK,\ \Gamma := \{x \mid x \in \{1,\ldots,U\}\})$:

   $t \xleftarrow{\mathsf{U}} \mathbb{Z}_p,\ K = g^\alpha g^{at},\ L = g^t,\ K_x = h_x^t,\ \forall x \in \Gamma,$

   return $sk_\Gamma := (K,\ L,\ \{K_x\}_{x\in\Gamma})$.

3. $\mathsf{Enc}(PK,\ m \in \mathbb{G}_T,\ \mathbb{A} := (M,\rho))$:

   Here, M is $\ell \times r$ matrix and $\rho : \{1,\ldots,\ell\} \to \{1,\ldots,U\}$,

   $\vec{v} = (s, y_2, \ldots, y_r) \xleftarrow{\mathsf{U}} \mathbb{Z}_p^r,\ C = me(g,g)^{\alpha s},\ C' = g^s,$

   $\lambda_i = \vec{v}\cdot M_i^T$ where M_i is i-th row of M for $i = 1,\ldots,\ell,$

   $C_i = g^{a\lambda_i} h_{\rho(i)}^{-s}$ for $i = 1,\ldots,\ell,$

   return $CT_{\mathbb{A}} := (C, C', \{C_i\}_{i=1}^{\ell}, \mathbb{A})$.

4. $\mathsf{Dec}(PK,\ sk_\Gamma,\ CT_{\mathbb{A}})$:

   If $\mathbb{A}$ accepts Γ, then compute I and $\{\omega_i\}_{i\in I}$ s.t. $\vec{T} = \sum_{i\in I} \omega_i M_i$,

   $I = \subseteq \{i \in \{1,\ldots,\ell\} \mid \rho(i) \in \Gamma\},$

   $K' = e(\prod_{i\in I} C_i^{-\omega_i}, L) \cdot e(C', K \prod_{i\in I} K_{\rho(i)}^{-\omega_i})$

   $= e(g,g)^{\alpha s} e(g,g)^{ast} e(g,g)^{-\prod_{i\in I} ta\lambda_i\omega_i}$

   $= e(g,g)^{\alpha s}$ (since $\sum_{i\in I} \omega_i\lambda_i = s$),

   return C/K'.

B FEIP

In this section, we explain the functional encryption for inner products (FEIP) proposed by Abdalla et al. [1] briefly.

B.1 Definition of FEIP

Definition 10 (Functional Encryption for Inner Products)**.** *A functional encryption for inner products (FEIP) scheme consists of the following algorithms. These are randomized algorithms except for* Dec*.*

1. $\mathsf{Setup}(1^\lambda, n)$

   Setup *algorithm takes as input a security parameter λ and the dimension of vector n. It outputs a pair of public parameter and master secret key (PK, MSK).*

2. KeyGen$(MSK, \vec{x})$

   KeyGen *takes as input a master secret key MSK and a vector* $\vec{x}$*. It outputs a private key* $sk_{\vec{x}}$*.*

3. Enc$(PK, \vec{y})$

   Enc *takes as inputs a public key PK and a vector* $\vec{y}$*. It outputs a ciphertext CT.*

4. Dec$(sk_{\vec{x}}, CT)$

   Dec *takes as inputs a private key* $sk_{\vec{x}}$ *for vector* $\vec{x}$ *and a ciphertext CT. It outputs an inner product value* $\vec{x} \cdot \vec{y}$ *where* $\vec{y}$ *is encrypted as CT and nothing else. (That is, a decryptor gets inner product value only, not any more information of* $\vec{y}$*.)*

B.2 Construction of FEIP from DDH

We explain the DDH assumption briefly. We define $\mathscr{G}_{\mathsf{gg}}$ as a probabilistic polynomial time algorithm that takes as input a security parameter 1^λ and outputs a triplet $(p, \mathbb{G}_p, g)$ where $\mathbb{G}_p$ is a group of order p which is a λ-bit prime number. g is a generator of $\mathbb{G}_p$, i.e., $g \in \mathbb{G}_p$. Then, the DDH assumption states that the tuples (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^z) are computationally indistinguishable, where $(p, \mathbb{G}_p, g) \xleftarrow{\mathsf{R}} \mathscr{G}_{\mathsf{gg}}(1^\lambda)$ and $a, b, z \xleftarrow{\mathsf{U}} \mathbb{Z}_p$. We show the construction of FEIP from DDH in [1] below:

1. Setup$(1^\lambda, n)$:

   $\mathsf{param} := (p, \mathbb{G}_p, g) \xleftarrow{\mathsf{R}} \mathscr{G}_{\mathsf{gg}}(1^\lambda)$, $\vec{s} := (s_1, \ldots, s_n) \xleftarrow{\mathsf{U}} \mathbb{Z}_p^n$,

   $PK := (\mathsf{param}, \{h_i = g^{s_i}\}_{i=1}^n)$, $MSK := \vec{s}$,

return (PK, MSK).

2. KeyGen$(MSK, \vec{x} := (x_1, \ldots, x_n) \in \mathbb{Z}_p^n)$:

return $SK_{\vec{x}} := \vec{x} \cdot \vec{s}$.

3. Enc$(PK, \vec{y} := (y_1, \ldots, y_n) \in \mathbb{Z}_p^n)$:

   $r \xleftarrow{\mathsf{U}} \mathbb{Z}_p, \mathsf{c}_0 := g^r, \mathsf{c}_i := h_i^r \cdot g^{y_i}$,

return $CT := (\mathsf{c}_0, \{\mathsf{c}_i\}_{i=1}^n)$.

4. Dec$(sk_{\vec{x}}, CT)$:

   $\mathsf{tmp} := \prod_{i=1}^n \mathsf{c}_i^{x_i} / \mathsf{c}_0^{sk_{\vec{x}}} = \prod_{i=1}^n (g^{s_i r + y_i})^{x_i} / g^{r \vec{x} \cdot \vec{s}} = g^{(\vec{x} \cdot \vec{s}) r + \vec{x} \cdot \vec{y} - r(\vec{x} \cdot \vec{s})} = g^{\vec{x} \cdot \vec{y}}$,

   compute and return the discrete logarithm of tmp to the base g.

C Security Proof

Here, let $\mathscr{C}$ and $\mathscr{B}$ be the challenger of q-DBDHE problem and the simulator of our proposed EABEFD (i.e., the adversary of q-DBDHE problem) respectively.

Theorem 1. *For any polynomial-time adversary of our proposed EABEFD* $\mathscr{A}$ *and any security parameter* λ*, the proposed EABEFD is restricted-selectively payload-hiding against chosen plaintext attacks under the q-DBDHE assumption and the following equation holds where the challenge access structure is* $\mathbb{A}^* := (M^*, \rho^*)$ *and* M^* *is an* $\ell^* \times r^*$ *access matrix and* $r^* \leq q$*:*

$$\mathsf{Adv}_{\mathscr{B}}^{\mathsf{q\text{-}DBDHE}}(1^\lambda) = \mathsf{Adv}_{\mathscr{A}}^{\mathsf{EABEFD,r\text{-}PH}}(1^\lambda)$$

Proof. We employ the proof of [26] to prove the security of our proposed scheme.

Init

$\mathscr{C}$ runs $\mathscr{B}$ and gives q-DBDHE challenge $(\mathsf{param}_{\mathbb{G}}, \vec{y}, T_b)$ to $\mathscr{B}$. $\mathscr{A}$ gives the challenge access structure $\mathbb{A}^* := (M^*, \rho^*)$ to $\mathscr{B}$. Here, we assume that M^* is an $\ell^* \times r^*$ access matrix and $r^* \leq q$.

Setup

$\mathscr{B}$ picks $\mathsf{param}_{\mathbb{G}}$ from q-DBDHE challenge. Then, $\mathscr{B}$ computes $\beta \xleftarrow{\cup} \mathbb{Z}_p^{\times}$, $s_{t,j} \xleftarrow{\cup} \mathbb{Z}_p^{\times}$ (for $t = 1,\ldots,d; j = 1,\ldots,n_t$) and generates $g^{\beta}, \{g^{s_{t,j}}\}_{t=1;j=1}^{d;n_t}$ by using g in $\mathsf{param}_{\mathbb{G}}$. Next, $\mathscr{B}$ chooses $\alpha' \xleftarrow{\cup} \mathbb{Z}_p$ and sets implicitly $\alpha = \alpha' + a^{q+1}$.Moreover, $\mathscr{B}$ generates $e(g,g)^{\alpha} = e(g^a, g^{a^q}) e(g,g)^{\alpha'}$ by using q-DBDHE challenge. In addition, for $t = 1,\ldots,d$, $\mathscr{B}$ computes $\gamma_t \xleftarrow{\cup} \mathbb{Z}_p$ and sets h_t as follows.: If $\rho^*(i') = (t, \vec{v}_{i'})$ for some $i' \in \{1,\ldots,\ell^*\}$, $\mathscr{B}$ sets $h_t = g^{\gamma_t} g^{aM^*_{i',1}} \cdot g^{a^2 M^*_{i',2}} \cdots g^{a^{r^*} M^*_{i',r^*}}$. Otherwise, $\mathscr{B}$ sets $h_t = g_t^{\gamma}$. Further, $\mathscr{B}$ gives $PK := (\mathsf{gparam} := (\mathsf{param}_{\mathbb{G}}, e(g,g)^{\alpha}, g^a, g^{\beta}), \{h_t\}_{t=1}^{d}, \{g^{s_{t,j}}\}_{t=1;j=1}^{d;n_t})$ to $\mathscr{A}$ and has $(\{\vec{s}_t := (s_{t,1},\ldots,s_{t,n_t})\}_{t=1}^{d}, \{\gamma_t\}_{t=1}^{d}, \beta)$ in secret.

Phase1

For $i = 1,\ldots,q_1$, $\mathscr{A}$ issues a key query for $\Gamma_i := \{(t, \vec{x}_t)\}$ to $\mathscr{B}$. Then, $\mathscr{B}$ generates user's secret key sk_{Γ_i} as follows. Here, we note that $\vec{T} \notin \mathsf{span}\langle (M^*_{i'})_{i' \in I'} \rangle$ for any Γ_i as described in **Phase** 1 of Definition 7.

$\mathscr{B}$ generates $\sigma \xleftarrow{\cup} \mathbb{Z}_p$, $\vec{w} = (w_1,\ldots,w_{r^*}) \in \mathbb{Z}_p^{r^*}$ s.t. $w_1 = -1$. Because Γ_i satisfies the restriction ($\vec{T} \notin \mathsf{span}\langle (M^*_{i'})_{i' \in I'} \rangle$ for any Γ_i), there exists such $\vec{w}$ satisfying $M^*_{i'} \cdot \vec{w}^T = 0$ for all $i' \in I'$. Next, $\mathscr{B}$ sets $z = \sigma + w_1 a^q + w_2 a^{q-1} + \cdots + w_{r^*} a^{q-r^*+1}$. Then, $\mathscr{B}$ also sets $L = g^{\sigma} \prod_{j'=1,\cdots,r^*} \left(g^{a^{q+1-j'}}\right)^{w_{j'}} = g^z$ and $K = g^{\alpha'} g^{a\sigma} \prod_{j'=2,\ldots,r^*} \left(g^{a^{q+2-j'}}\right)^{w_{j'}} = g^{\alpha} g^{az}$.

Moreover, for any t s.t. $(t, \vec{x}_t) \in \Gamma_i$, $\mathscr{B}$ sets K_t as follows.: If $\rho^*(i') = (t, \vec{v}_{i'})$ for some $i' \in \{1,\ldots,\ell^*\}$, $\mathscr{B}$ sets $K_t = L^{\gamma_t} \prod_{j'=1,\ldots,r^*} (g^{a^{j'} \cdot \sigma} \prod_{k=1,\ldots,r^*;k\neq j'} \left(g^{a^{q+1+j'-k}}\right)^{w_k})^{M^*_{i',j'}} = h_t^z$.

Otherwise, $\mathscr{B}$ sets $K_t = L^{\gamma_t}$.

After that, for any $(t, \vec{x}_t) \in \Gamma_i$, $\mathscr{B}$ sets

$\tau_{(t,\vec{x}_t)} \xleftarrow{\cup} \mathbb{Z}_p^{\times}$,

$k_{(t,\vec{x}_t)} = \tau_{(t,\vec{x}_t)} \vec{x}_t \cdot \vec{s}_t$,

$\tilde{k}_{(t,\vec{x}_t)} = L^{(1-\tau_{(t,\vec{x}_t)})(\sum_{j=1}^{n_t} x_{t,j} s_{t,j})/\beta} = (g^{\sigma} \prod_{j'=1,\cdots,r^*} \left(g^{a^{q+1-j'}}\right)^{w_{j'}})^{(1-\tau_{(t,\vec{x}_t)})(\sum_{j=1}^{n_t} x_{t,j} s_{t,j})/\beta}$

$= g^{z(1-\tau_{(t,\vec{x}_t)})\vec{x}_t \cdot \vec{s}_t/\beta}$

and generates $k_{(t,\vec{x}_t)}, \tilde{k}_{(t,\vec{x}_t)}$. $\mathscr{B}$ gives $sk_{\Gamma_i} := (\Gamma_i, K, L, \{K_t\}_{t \text{ s.t. } (t,\vec{x}_t) \in \Gamma_i}, \{k_{(t,\vec{x}_t)}, \tilde{k}_{(t,\vec{x}_t)}\}_{(t,\vec{x}_t) \in \Gamma_i})$ to $\mathscr{A}$.

Remark: If we try to prove that our scheme achieves selective security (not restricted-selective security), we cannot prove it because we cannot generate $\vec{w}$. The reason why we cannot generate $\vec{w}$ is the diffenrence of map ρ^* between our scheme and previous work [26]. In our scheme, ρ^* is the map $\{1,\ldots,\ell\} \to \{(t,\vec{v}), \ldots, \neg(t,\vec{v}),\ldots\}$. On the other hand, in [26], ρ^* is the map $\{1,\ldots,\ell\} \to \{1, \ldots, U\}$. That is, ρ^* in our scheme associates the row of challenge access matrix with the pair of category and positive/negative attribute vector. ρ^* in [26] associates the row of challenge access matrix with the attribute.

For example, we assume that there are attribute categories t_1 and t_2 in the ABE system. t_1 includes attribute vectors $\vec{a}_1$ and $\vec{b}_1$ (which are associated with attributes a_1 and b_1 respectively). t_2 includes attribute vectors $\vec{a}_2$ and $\vec{b}_2$ (which are associated with attributes a_2 and b_2 respectively). We also assume the challenge access policy is $a_1 \wedge \neg a_2$. In the selective security game, the adversary can

query the private key for $\{a_1, a_2\}$ because the set of attributes does not satisfy $\mathbb{A}^*$ and the key query restriction is satisfied. However, if the adversary can query the private key for $\{(t_1,\vec{a}_1),\ (t_2,\vec{a}_2)\}$ in our construction, $\vec{w}$ (s.t. $\forall i', M^*_{i'} \cdot \vec{w}^T = 0$ where $\rho^*(i') \in S$) does not exist except $\vec{0}$. That is, Water's proof methodology in [26] cannot be employed in proving our scheme achieves selective security. Therefore, we need to introduce a new security notion, restricted-selective security, and the stronger key query restriction than one in selective security.

Challenge

$\mathscr{B}$ computes $C = m^*_\mu T_b e(g^s, g^{\alpha'})$, $C' = g^s$ and generates $r' \xleftarrow{\mathsf{U}} \mathbb{Z}_p^\times$, $c = g^{r'}$ and $\tilde{c} = g^{\beta r'}$.

Moreover, $\mathscr{B}$ picks ${y'}_2, \ldots, {y'}_{r^*} \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ and sets $\vec{f}$ as follows. (Here, ${y'}_1 = 0$.):

$$\vec{f} = (s, sa + {y'}_2, sa^2 + {y'}_3, \ldots, sa^{r^*-1} + {y'}_{r^*}) \in \mathbb{Z}_p^{r^*},$$

For $i' = 1, \ldots, \ell^*$, $\mathscr{B}$ generates $\theta_{i'} \xleftarrow{\mathsf{U}} \mathbb{Z}_p^\times$ and defines $\{c_{i',j}\}_{i'=1;j=1}^{\ell^*;n_t}$ as follows:

if $\rho^*(i') = (t, \vec{v}_{i'} := (v_{i',1}, \ldots, v_{i',n_t}) \in \mathbb{Z}_p^{n_t} \setminus \{\vec{0}\})$,

$$c_{i',1} := g^{r' s_{t,1}} \Big(\prod_{j'=1}^{r^*} (g^a)^{M^*_{i',j'} y'_{j'}}\Big)(g^s)^{-\gamma_t} g^{\theta_{i'} v_{i',1}},\ c_{i',2} := g^{r' s_{t,2}} g^{\theta_{i'} v_{i',2}},\ \ldots,\ c_{i',n_t} := g^{r' s_{t,n_t}} g^{\theta_{i'} v_{i',n_t}},$$

if $\rho^*(i') = \neg(t, \vec{v}_{i'})$,

$$c_{i',1} := g^{r' s_{t,1}} \Big(\Big(\prod_{j'=1}^{r^*} (g^a)^{M^*_{i',j'} y'_{j'}}\Big)(g^s)^{-\gamma_t}\Big)^{v_{i',1}},$$

$$c_{i',2} := g^{r' s_{t,2}} \Big(\Big(\prod_{j'=1}^{r^*} (g^a)^{M^*_{i',j'} y'_{j'}}\Big)(g^s)^{-\gamma_t}\Big)^{v_{i',2}},\ \ldots\ c_{i',n_t} := g^{r' s_{t,n_t}} \Big(\Big(\prod_{j'=1}^{r^*} (g^a)^{M^*_{i',j'} y'_{j'}}\Big)(g^s)^{-\gamma_t}\Big)^{v_{i',n_t}},$$

After that, $\mathscr{B}$ gives $CT_{\mathbb{A}^*} := (\mathbb{A}^*, C, C', c, \tilde{c}, \{c_{i',j}\}_{i'=1;j=1}^{\ell;n_t})$ to $\mathscr{A}$.

Phase2

For $i = 1, \ldots, \nu$, $\mathscr{A}$ issues key query for $\Gamma_i := \{(t, \vec{x}_t)\}$ to $\mathscr{B}$. $\mathscr{B}$ generates user's secret keys as in **Phase1**. However, it holds that $\vec{T} \notin \mathsf{span}\langle (M^*_{i'})_{i' \in I'} \rangle$ for any Γ_i.

Guess

$\mathscr{A}$ guesses μ and outputs μ'. Then, $\mathscr{A}$ gives μ' to $\mathscr{B}$. If $\mu = \mu'$, then $\mathscr{B}$ outputs $b' = 0$ and gives it to $\mathscr{C}$. Otherwise, $\mathscr{B}$ outputs $b' = 1$ and gives it to $\mathscr{C}$.

Here, we have the following:

$$\begin{aligned}
&\mathsf{Adv}^{\text{q-DBDHE}}_{\mathscr{B}}(\lambda) \\
&= |\Pr[b' = 0 \mid b = 0] - \Pr[b' = 0 \mid b = 1]| \\
&= |\Pr[\mu = \mu' \mid b = 0] - \Pr[\mu \neq \mu' \mid b = 1]| \\
&= |\Pr[\mu = \mu' \mid \mathscr{A}\ \textit{in}\ \mathsf{EABEFD}\textit{'s Game}] - 1/2| \\
&= \mathsf{Adv}^{\mathsf{EABEFD,r\text{-}PH}}_{\mathscr{A}}(\lambda)
\end{aligned}$$

Furthermore, q-DBDHE assumption holds and $r^* \leq q$, and thus $\mathsf{Adv}^{\mathsf{EABEFD,r\text{-}PH}}_{\mathscr{A}}(\lambda)$ is negligible.

Hence, Theorem 1 holds.

□

Author Biography

Hikaru Tsuchida receiverd the B.S. and M.S. degrees in Information Science from Tokyo University of Science and in Risk Engineering from University of Tsukuba in 2014 and 2016, respectively. He is now with NEC Corporation.

Takashi Nishide received his B.S. degree from the University of Tokyo in 1997, M.S. degree from the University of Southern California in 2003, and Dr.E. degree from the University of Electro-Communications in 2008. From 1997 to 2009, he had worked at Hitachi Software Engineering Co., Ltd., developing security products. From 2009 to 2013, he had been an assistant professor at Kyushu University and from 2013 he is an associate professor at University of Tsukuba. His research is in the areas of cryptography and information security.

Eiji Okamoto received his B.S., M.S. and Ph.D. degrees in electronics engineering from Tokyo Institute of Technology in 1973, 1975 and 1978, respectively. He worked and studied communication theory and cryptography for NEC central research laboratories since 1978. From 1991 he became a professor at Japan Advanced Institute of Science and Technology, Toho University, and University of Tsukuba. He is a professor emeritus at University of Tsukuba now. His research interests are cryptography and information security. He is members of IEEE and ACM.