Revocable Decentralized Multi-Authority Functional Encryption

Hikaru Tsuchida[1(✉)], Takashi Nishide[2], Eiji Okamoto[2], and Kwangjo Kim[3]

[1] NEC Corporation, 1753, Shimonumabe, Nakahara-Ku,
Kawasaki, Kanagawa 211-8666, Japan
h-tsuchida@bk.jp.nec.com

[2] Faculty of Engineering, Information and Systems, University of Tsukuba,
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
{nishide,okamoto}@risk.tsukuba.ac.jp

[3] Computer Science Department, KAIST, 291 Daehak-ro, Yuseong-gu,
Daejeon 305-701, Korea
kkj@kaist.ac.kr

Abstract. Attribute-Based Encryption (ABE) is regarded as one of the most desirable cryptosystems realizing data security in the cloud storage systems. Functional Encryption (FE) which includes ABE and the ABE system with multiple authorities are studied actively today. However, ABE has the attribute revocation problem. In this paper, we propose a new revocation scheme using update information, i.e., revocation patch (not update key), in which an encryptor does not need to care about the revocation list. We propose an FE scheme with multiple authorities and no central authority supporting revocation by using revocation patch. Our proposal realizes the revocation on the attribute level. More precisely, we introduce the new concept, i.e., the revocation on the category level that is a generalization of attribute level. We prove that our construction is adaptively secure against chosen plaintext attacks and static corruption of authorities based on the decisional linear (DLIN) assumption.

Keywords: Functional encryption · Access control · Multiple authorities · Revocation · Attribute-level

1 Introduction

1.1 Background

In recent years, outsourcing data storage to cloud service providers has been increasing. Due to this change, there are frequent leaks of confidential data in cloud storage system. Therefore, data security in the cloud server is required. Attribute-Based Encryption (ABE) [3,6,9,13,14,19,22] is regarded as one of the

This work was completed while the corresponding author was a graduate student at University of Tsukuba.

O. Dunkelman and S.K. Sanadhya (Eds.): INDOCRYPT 2016, LNCS 10095, pp. 248–265, 2016.
DOI: 10.1007/978-3-319-49890-4_14

most desirable cryptosystems realizing data security in the cloud storage systems. ABE systems can provide data security and access control without a trusted server by using access policies and associated attributes among ciphertexts and private keys. For example, if the data owner encrypts data with an access policy like ("the sales department" OR ("the development department" AND "chief")), only a staff member in the sales department and a chief of the development department can decrypt the data in a Ciphertext-Policy ABE (CP-ABE) system. Furthermore, Functional Encryption (FE) [5,16,18] which includes ABE and the ABE system with multiple authorities [6,13,14,18] are proposed.

However, CP-ABE has the attribute revocation problem. For example, if a staff member in the sales department got fired and still has a decryption key related to the attribute of "the sales department" illegally, he/she may be able to decrypt encrypted data associated with an access policy related to "the sales department". Accordingly, a CP-ABE system needs a user (attribute) revocation scheme. In previous research, there are two types of revocation schemes: indirect revocation [4,11,21] and direct revocation [1]. The former scheme requires an update key for revocation issued by an authority, but an encryptor does not have to care about a revocation list in the indirect revocation system. The latter scheme can revoke users without using an update key because an encryptor specifies revoked users for ciphertexts by using the revocation list which may be specified and given by the authority or may be specified freely by the encryptor. In other words, in the direct revocation system, an encryptor has to care about a revocation list. That is, in indirect (direct) revocation, users are revoked by an authority (encryptor resp.). The direct revocation with multiple authorities was already proposed in [10], but the indirect one is not proposed.

To achieve attribute-level revocation, encryptors will expect that individual authorities such as universities and government maintain revocation lists rather than specifying the revocation list for ciphertexts by themselves. Furthermore, it is also desirable that a new cryptosystem has expressiveness of access policies (e.g. an FE system) and practical attribute management (e.g. multi-authority ABE system). For this reason, we propose FE with multiple authorities supporting indirect revocation, i.e., an encryptor does not have to care about revocation list.

1.2 Our Results

We propose a new revocation scheme using what we call revocation patch, *patch revocation scheme*, by combining indirect revocation and Decentralized Multi-Authority Functional Encryption (DMA-FE) [18] (which realizes non-monotone access structures using inner-product relations with multiple authorities and no central authority), i.e. the first DMA-FE scheme supporting indirect revocation. Our proposed scheme realizes the revocation on the category level that is a generalization of indirect revocation on the attribute level. (For more information about revocation on the attribute level, see the full version of this paper [23].)

Here, we give the intuitive explanation of how to specify revocation with a toy example of revocation on the category level in our scheme. Suppose that, the attribute category t_1 includes the attribute values $A_{t_1}, A'_{t_1}, A''_{t_1}$ and the attribute category t_2 includes the attribute values $B_{t_2}, B'_{t_2}, B''_{t_2}$. We would like to specify $(\neg A_{t_1} \wedge \neg B_{t_2})$ as a ciphertext policy as described below. Then, we specify revocation information in the ciphertext on the category level, as, for example, $((\neg A_{t_1} \wedge t_1[\mathsf{v}_{t_1}]) \wedge (\neg B_{t_2} \wedge t_2[\mathsf{v}_{t_2}]))$. Here, $(\neg A_{t_1} \wedge t_1[\mathsf{v}_{t_1}])$ means that the decryptor needs to have attribute of t_1 except A_{t_1}, i.e., needs to have A'_{t_1} or A''_{t_1}. Moreover, the decryptor's attribute of t_1 needs to be not revoked before issuing the revocation patch of version v_{t_1} by the authority that manages attribute category t_1. $(\neg B_{t_2} \wedge t_2[\mathsf{v}_{t_2}])$ means the similar condition about t_2 as for t_1. If we would like to specify a non-monotone access structure for a ciphertext, the revocation information is required to be on the category level, not attribute level. The revocation on the attribute level considers that the user's attribute which is associated with an access policy is valid or revoked, but does not consider the other attributes of the category to which the user's attribute belongs. For this reason, our scheme supporting non-monotone access control realizes the revocation on the category level, rather than attribute level. We also prove that our construction is adaptively secure against chosen plaintext attacks and static corruption of authorities based on the DLIN assumption. (We note that DMA-FE [18] of Okamoto and Takashima does not achieve the security against static corruption of authorities.)

We show a comparison with previous works in Tables 1 and 2. In Table 1, LSSS means Linear Secret Sharing Scheme. In Table 2, SD method means Subset Difference method in [15]. Std. model, GBGM and ROM mean standard

Table 1. Comparison with previous works

Schemes	Authority (central authority)	Policy	Access structure
AI09 [2]	Single	Key-Policy	Monotone (LSSS)
DDM15 [7]	Single	Key-Policy	Non-monotone (LSSS)
H15 [10]	Multiple (Y)	Ciphertext-Policy	Monotone (LSSS)
L12 [13]	Multiple (N)	Ciphertext-Policy	Monotone (LSSS)
OT13 [18]	Multiple (N)	Ciphertext-Policy	Non-monotone (LSSS & Inner-Product)
This work	Multiple (N)	Ciphertext-Policy	Non-monotone (LSSS & Inner-Product)

Table 2. Comparison with previous works (cont.)

Schemes	Revocation	Revocation level	Security model	Assumption
AI09 [2]	Direct/Indirect (CS method)	User level	selective (Std. model)	DBDH
DDM15 [7]	Direct (SD method)	User level	full (Std. model)	DLIN
H15 [10]	Direct	User level	full$^+$ (GBGM & ROM)	-
L12 [13]	-	-	full$^+$ (ROM)	DLIN
OT13 [18]	-	-	full (ROM)	DLIN
This work	Patch (CS method)	Category level	full$^+$ (ROM)	DLIN

model, generic bilinear group model and random oracle model, respectively. The "full" and "full^{+}" mean "adaptively payload-hiding against chosen plaintext attacks" and "adaptively payload-hiding against chosen plaintext attacks and static corruption of authorities". DBDH means the Decisional Bilinear Diffie-Hellman assumption. Tables 1 and 2 show that our proposal has expressiveness of access policy and practical attribute management. It also shows that our proposed scheme realizes that each of the authorities is able to revoke user's attribute by themselves (not an encryptor).

1.3 Key Techniques

Overview. Our scheme is based on DMA-FE [18]. In DMA-FE [18], there are roughly two types of ciphertexts: the encrypted message and the headers for access control. Only the user who has attribute keys associated with the access policy can restore the secret (or session key) from the headers and decrypt the encrypted message by using it. We add keys and headers for attribute revocation to DMA-FE [18] by introducing the basis of dual pairing vector spaces (DPVS) for attribute revocation. Due to this, only the user who has attribute keys associated with the access policy and keys for attribute revocation which are not revoked can get the message. A key for attribute revocation is like an attribute key in DMA-FE [18], so an attribute vector is embedded in a key for attribute revocation. Each key for attribute revocation is tied to every attribute key. Therefore, if the key for attribute revocation is revoked, the attribute key to which it is tied also becomes the invalid key.

How to Revoke. We realize a mechanism that we call patch that provides the same functionality as indirect revocation by using DPVS and devising the encoding of an attribute vector like a full binary tree. In the patch revocation scheme, an attribute authority prepares a full binary tree of users for every attribute category and issues the latest revocation patch associated with a covering node of the full binary tree (by running $\mathsf{PUpdate}$ algorithm) when the event of user's attribute revocation occurred. The revocation patch is the update information and equivalent to update keys in indirect revocation. Issuing the latest revocation patches of each attribute by each attribute authority realizes the revocation on the category level. When an encryptor generates the ciphertext with the access policy, he/she obtains the latest revocation patches of each attribute associated with the access policy and applies it to the ciphertext, i.e., makes the headers for attribute revocation by using the latest revocation patches. If the product of attribute vector (which represents the user's label) in the key for attribute revocation and the header's attribute vector (which represents the path of the covering node) is not zero, the key for attribute revocation is revoked.

Comparison Between Patch Revocation and Indirect Revocation. If there are many decryptors, the patch revocation scheme is superior to the indirect revocation scheme because the patch revocation scheme can reduce the communication cost and process of decryptors. For details of comparison between patch revocation and indirect revocation, see the full version of this paper [23].

How to Prove Security. We employ the Dual System Encryption (DSE) methodology in [18] to prove the adaptive security. However, we cannot apply the DSE directly because the attacker of our proposal can request user's attribute keys and keys for attribute revocation that satisfy the challenge access structure (but some user's private key is revoked). That is, we cannot use the key query restriction in the security proof straightforwardly. To solve this problem, we use the secret sharing and the proof methodology in [7]. Furthermore, to prove the security against the static corruption of authorities, we use the technique in [13].

1.4 Related Works

ABE (with Single Authority): Sahai and Waters introduced Fuzzy Identity-Based Encryption (FIBE) [22] that is a special type of ABE. The only access structure supported in FIBE is "threshold". In FIBE, ciphertexts and user's private key are associated with a set of attributes ω and both a threshold parameter d and another set of attributes ω', respectively. Then, if $|\omega \cap \omega'| \geq d$ holds, the user can decrypt ciphertexts and get the plaintext. Some ABE is studied and developed actively after [22] is introduced. ABE can provide data security and access control without a trusted server by using access policies and associated attributes among ciphertexts and user's private keys. Key-Policy ABE (KP-ABE) [9] introduced by Goyal et al. is the scheme that supports an access structure in user's private key. CP-ABE [3] introduced by Bethencourt et al. is the scheme that supports an access structure in ciphertexts. Ostrovsky et al. [19] proposed a scheme that supports non-monotone access structure where negated attributes are available. In recent years, FE [5,16] including ABE is proposed.

Multi-Authority ABE: Chase proposed the first multi-authority ABE [6] that extends FIBE. After that, Müller et al. proposed the multi-authority ABE [14] that extends CP-ABE. In recent years, Lewko proposed an adaptively secure multi-authority CP-ABE against static corruption of authorities [12,13] and Okamoto and Takashima proposed multi-authority functional encryption without a central authority (DMA-FE) [18].

Revocation: Boldyreva et al. introduced the IBE supporting revocation by update key [4]. After that, Sahai et al. proposed the ABE supporting revocation by update keys and updating ciphertext [21]. Recently, Lee et al. introduced a new cryptographic primitive realizing a time-evolution mechanism [11], in other words, Lee et al. proposed a new revocation scheme with modularity. Meanwhile, Attrapadung et al. proposed the ABE supporting revocation without update keys which specifies revoked users for ciphertexts directly [1]. Attrapadung et al. also proposed the ABE supporting (user-level) direct/indirect revocation [2]. Qian et al. proposed the KP-ABE supporting direct revocation and achieving adaptive security in composite order bilinear groups [20]. Gonz*á*lez-Nieto et al. proposed the full-hiding revocable predicate encryption supporting direct revocation where the revocation list is hidden specified for ciphertexts [8][1]. Datta et al. proposed the (unbounded) KP-ABE supporting direct revocation by using a subset difference method in prime order bilinear groups [7]. Horváth proposed multi-authority ABE (with a central authority) which specifies revoked users for ciphertexts directly [10].

1.5 Notations

We follow the notations in [11,18].

When A is a random variable or distribution, $y \xleftarrow{\mathsf{R}} A$ denotes that y is randomly selected from A according to its distribution. When A is a set, $y \xleftarrow{\mathsf{U}} A$ denotes that y is uniformly selected from A. We denote the finite field of order q by $\mathbb{F}_q$, and $\mathbb{F}_q \setminus \{0\}$ by $\mathbb{F}_q^\times$. A vector symbol denotes a vector representation over $\mathbb{F}_q$, e.g., $\vec{x}$ denotes $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$. For two vectors $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{v} = (v_1, \ldots, v_n)$, $\vec{x} \cdot \vec{v}$ denotes the inner-product $\sum_{i=1}^n x_i v_i$. The vector $\vec{0}$ is abused as the zero vector in $\mathbb{F}_q^n$ for any n. X^T denotes the transpose of matrix X. A bold face letter denotes an element of vector space $\mathbb{V}$, e.g., $\boldsymbol{x} \in \mathbb{V}$. When $\boldsymbol{b}_i \in \mathbb{V} (i = 1, \ldots, n), \mathsf{span}\langle \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \rangle \subseteq \mathbb{V}$ (resp. $\mathsf{span}\langle \vec{x}_1, \ldots, \vec{x}_n \rangle$) denotes the subspace generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ (resp. $\vec{x}_1, \ldots, \vec{x}_n$). For bases $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$ and $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$, $(x_1, \ldots, x_N)_{\mathbb{B}} := \sum_{i=1}^N x_i \boldsymbol{b}_i$ and $(y_1, \ldots y_N)_{\mathbb{B}^*} := \sum_{i=1}^N y_i \boldsymbol{b}_i^*$. For a format of attribute vectors $\vec{n} := (d; n_{A,1}, \ldots, n_{A,d}, n_{R,1}, \ldots, n_{R,d})$ that indicates dimensions of vector spaces, $\vec{e}_{\mathsf{f},t,j}$ denotes the canonical basis vector $(\overbrace{0, \ldots, 0}^{j-1}, 1, \overbrace{0, \ldots, 0}^{n_{\mathsf{f},t}-j}) \in \mathbb{F}_q^{n_{\mathsf{f},t}}$ for $\mathsf{f} = A, R; t = 1, \ldots, d; j = 1, \ldots, n_{\mathsf{f},t}$, where f and t represent the functionality (A represents the access control and R represents the revocation) and the attribute authority. $GL(n, \mathbb{F}_q)$ denotes the general linear group of degree n over $\mathbb{F}_q$.

For a string $L \in \{0,1\}^n$, let $L[i]$ be the ith bit of, L, and $L|_i$ be the prefix of L with $i-$bit length. For example, if $L = 010$, then $L[0] = *, L[1] = 0, L[2] = 1, L[3] = 0$, and $L|_0 = *, L|_1 = 0, L|_2 = 01, L|_3 = 010$.

1.6 Preliminaries

We use DPVS introduced by Okamoto and Takashima [16] and general predicates (non-monotone access structures using inner-product relations). We also use the subset-cover revocation framework introduced by Naor et al. [15]. For these preliminaries, see the full version of this paper [23].

[1] The scheme of [8] can hide the revocation list (i.e., identities of revoked users) specified for ciphertexts in a provably secure way, but an encryptor needs to care about revocation lists. We note that an encryptor does not have to care about the revocation list in the schemes supporting indirect revocation [4,11,21] and our scheme. However, we note that the aim of the indirect revocation [4,11,21] and our scheme is not to hide the revocation list specified for ciphertexts in a provably secure way.

2 Revocable Decentralized Multi-Authority Functional Encryption (R-DMA-FE)

2.1 Definitions of R-DMA-FE

Definition 1 (Revocable Decentralized Multi-Authority Functional Encryption). *A revocable decentralized multi-authority functional encryption (R-DMA-FE) scheme consists of the following algorithms. These are randomized algorithms except for* Dec*.*

1. $\mathsf{GSetup}(1^\lambda)$
   The GSetup *algorithm takes as input a security parameter* λ *and outputs a global parameter* gparam*.*
2. $\mathsf{ASetup}(\mathsf{gparam}, t, n_{A,t}, N_{max,t}, \varphi_t)$
   The ASetup *algorithm takes as input a global parameter* gparam*, an attribute authority (or category)* t $(1 \leq t \leq d)$*, a dimension of attribute vector space* $n_{A,t}$*, the maximum number* $N_{max,t}$ *of users for the attribute in the category* t *and the upper bound* φ_t *for the number of subsets in the cover. It outputs an attribute-authority public key* apk_t*, an attribute-authority secret key* ask_t*, an revocation public key* rpk_t *and an revocation secret key* rsk_t*.*
3. $\mathsf{PUpdate}(t, \mathsf{rpk}_t, \mathsf{rsk}_t, r\ell_{\mathsf{v}_t}, \mathsf{v}_t)$
   The $\mathsf{PUpdate}$ *takes as input an attribute authority (or category)* t*, a revocation public key* rpk_t*, a revocation secret key* rsk_t*, the latest revocation list* $r\ell_{\mathsf{v}_t}$[2] *and the latest version number for the revocation patch* v_t*. It outputs the latest revocation patch* $\mathsf{CP}_{\mathsf{v}_t}$*.*
4. $\mathsf{KeyGen}(\mathsf{gparam}, t, \mathsf{ask}_t, \mathsf{rsk}_t, \mathsf{gid}, \vec{x}_{A,t})$
   The KeyGen *takes as input a global parameter* gparam*, an attribute authority (or category)* t*, a revocation secret key* rsk_t*, the user* gid *and an attribute vector* $\vec{x}_{A,t}$*. It outputs the user secret key* $\mathsf{usk}_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}$ *where* rt *represents the number of return (after* gid*'s* $(t, \vec{x}_{A,t})$ *revocation*[3]*).*
5. $\mathsf{Enc}(\{\mathsf{apk}_t, \mathsf{rpk}_t, \mathsf{CP}_{\mathsf{v}_t}\}, m,\ \mathbb{S})$
   The Enc *takes as inputs a set of public keys from relevant authorities* $\{\mathsf{apk}_t, \mathsf{rpk}_t\}$*, a set of the latest revocation patches from relevant authorities* $\{\mathsf{CP}_{\mathsf{v}_t}\}$*, a message* $m \in \mathbb{G}_T$*, and an access structure* $\mathbb{S}$*. It outputs a ciphertext* $\mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}}$*.*
6. $\mathsf{Dec}(\mathsf{gparam}, \{\mathsf{apk}_t, \mathsf{rpk}_t, \mathsf{usk}_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}\}, \mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}})$
   The Dec *takes as inputs a set of public keys from relevant authorities* $\{\mathsf{apk}_t, \mathsf{rpk}_t\}$ *and secret keys* $\{\mathsf{usk}_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}\}$ *corresponding to user* gid *and pair of attributes and number of return after revocation* $\{((t, \vec{x}_{A,t}), \mathsf{rt})\}$ *and a ciphertext* $\mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}}$*. It outputs a message* m *or a special symbol* $\perp$*.*

An R-DMA-FE scheme should have the following correctness property: for all security parameter λ*, all attribute sets* $\Gamma := \{(t, \vec{x}_{A,t})\}$*, all* gid*, all the number of return (after* gid*'s* $(t, \vec{x}_{t,A})$ *revocation)* rt*, all messages* m*, all*

[2] We define a user's attribute revocation list with its version $\mathsf{v_t}$: $r\ell_{\mathsf{v_t}} \subseteq \{1, \ldots, N_{max,t}\}$.

[3] We assume that a revoked user can become unrevoked again (possibly several times) after the user was revoked.

access structures $\mathbb{S}$ *and all the latest revocation lists* $r\ell_{\mathsf{v}_t}$*, it holds that* $m = \mathsf{Dec}(\mathsf{gparam}, \{\mathsf{apk}_t, \mathsf{rpk}_t, \mathsf{usk}_{\mathsf{gid},(t,\vec{x}_{A,t})\in\Gamma,\mathsf{rt}}\}, \mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}})$ *with overwhelming probability, if* $\mathbb{S}$ *accepts* Γ *and* $\forall\delta$ *related with* Γ*, i.e.,* $\vec{1} \in \mathsf{span}\langle M_\delta\rangle$ *s.t.* $M_\delta := (M_j)_{\gamma(j)=1}$*, there exists no* j *s.t.* $\mathsf{FindNode}(\mathsf{gid}_i, (t, \vec{x}_{A,t}), \mathsf{rt}) \in r\ell_{\mathsf{v}_t} \in \{r\ell_{\mathsf{v}_t}\}_t$[4] *and* $\rho(j) = (t, \vec{x}_{A,t})$ *or* $\neg(t, \vec{x}_{A,t})$*, where*

$$\mathsf{gparam} \xleftarrow{\mathsf{R}} \mathsf{GSetup}(1^\lambda),$$

$$(\mathsf{apk}_t, \mathsf{ask}_t, \mathsf{rpk}_t, \mathsf{rsk}_t) \xleftarrow{\mathsf{R}} \mathsf{ASetup}(\mathsf{gparam}, t, n_{A,t}, N_{max,t}, \varphi_t),$$

$$\mathsf{CP}_{\mathsf{v}_t} \xleftarrow{\mathsf{R}} \mathsf{PUpdate}(t, \mathsf{rpk}_t, \mathsf{rsk}_t, r\ell_{\mathsf{v}_t}, \mathsf{v}_t),$$

$$\mathsf{usk}_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}} \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{gparam}, t, \mathsf{ask}_t, \mathsf{rsk}_t, \mathsf{gid}, \vec{x}_{A,t}),$$

$$\mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}} \xleftarrow{\mathsf{R}} \mathsf{Enc}(\{\mathsf{apk}_t, \mathsf{rpk}_t, \mathsf{CP}_{\mathsf{v}_t}\}, m, \mathbb{S}),$$

We let $\mathcal{S}$ *be the set of authorities. We assume each attribute is assigned to one authority and an attribute is considered to be of the form of* $(t, \vec{x}_t)$*. For simplicity, we also assume that each authority manages only one attribute category.*[5]

Definition 2 (Security of R-DMA-FE). *For an adversary, we define* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{R-DMA-FE,PHCA}}(\lambda)$ *to be the advantage in the following experiment for any security parameter* λ*. An R-DMA-FE scheme is adaptively payload-hiding secure against chosen plaintext attacks and static corruption of authorities if the advantage of any polynomial-time adversary is negligible:*

Setup

Given 1^λ*, the challenger gives* $\mathsf{gparam} \xleftarrow{\mathsf{R}} \mathsf{GSetup}(1^\lambda)$ *to adversary* $\mathcal{A}$*.* $\mathcal{A}$ *specifies a set* $\mathcal{S}' \subset \mathcal{S}$ *of corrupt authorities, where* $\mathcal{S}(:= \{1, \ldots, d\})$ *is the set of all the authorities in the system. For good authority* $t \in \mathcal{S} \setminus \mathcal{S}'$*, the challenger runs* $(\mathsf{apk}_t, \mathsf{ask}_t, \mathsf{rpk}_t, \mathsf{rsk}_t) \xleftarrow{\mathsf{R}} \mathsf{ASetup}(\mathsf{gparam}, t, n_{A,t}, N_{max,t}, \varphi_t)$ *and gives* $\{\mathsf{apk}_t, \mathsf{rpk}_t\}_{t\in\mathcal{S}\setminus\mathcal{S}'}$ *to* $\mathcal{A}$*.*

Phase 1

The adversary is allowed to issue a polynomial number of queries, $(\mathsf{gid}, (t, \vec{x}_{A,t}))$*, to the challenger or oracle* $\mathsf{KeyGen}(\mathsf{gparam}, t, \mathsf{ask}_t, \mathsf{rsk}_t, \cdot, \cdot)$ *for private keys, attribute secret key* $\mathsf{usk}_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}$*, where* t *is an attribute category belonging to a good authority,* gid *is an global identifier and* rt *is the number of return after* gid*'s* $(t, \vec{x}_{A,t})$ *revocation.*[6] *The adversary is also allowed*

[4] Here, we define $\mathsf{FindNode} : \{0,1\}^* \times \{(t, \vec{x}_{A,t})\} \times \mathbb{N} \cup \{0\} \rightarrow \{1, \ldots, N_{max,t}\}$. The $\mathsf{FindNode}$ is not a priori function. An attribute authority assigns $(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})$ to the $\mathsf{FindNode}(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})$-th leaf node newly and uniquely every time the user key is issued. We remark that an attribute authority can decide how to choose a leaf by itself as long as the assignment is unique. Then, let "user u" in the subset-cover revocation framework equal $\mathsf{FindNode}(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})$. That is, $\mathsf{FindNode}(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt}) = u \in \{1, \ldots, N_{max,t}\}$.

[5] We note that actually each authority can manage several attribute categories.

[6] For example, a user is initially unrevoked, and the user may be revoked. If the user becomes unrevoked again, then rt is 1.

to issue a polynomial number of queries, $(\{(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})\}_{t\in\mathcal{S}\setminus\mathcal{S}'}, \mathsf{v}_t)$*, to the challenger or oracle* $\mathsf{PUpdate}(t, \mathsf{rpk}_t, \mathsf{rsk}_t, \{\mathsf{FindNode}(\cdot,\cdot,\cdot)\}, \cdot)$ *for revocation patch* $\mathsf{CP}_{\mathsf{v}_t}$*. Note that the adversary is allowed to query only one revocation patch for each* t *and* v_t*.*

Challenge

Let $\Gamma_{\mathsf{gid}_i} := \{(t, \vec{x}_{A,t})\}(i = 1, \ldots, \nu)$ *be the queries set to the* KeyGen *oracle with* gid_i*. The adversary submits two messages* $m^{(0)}, m^{(1)}$*, an access structure* $\mathbb{S} := (M, \rho)$ *and the pair of revocation lists for relevant good authorities and the number of version* $\{(RL_t, \mathsf{v}_t) \mid RL_t := \{(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})\}\}_{t\in\mathcal{S}\setminus\mathcal{S}'}$*. We note that for a valid matrix (i.e., matrix used to specify an access structure by using a linear secret sharing scheme) in the security game, the rows associated with corrupt authorities cannot include the target vector* $\vec{1}$ *in their span. The access structure and revocation history must satisfy at least one of the following restrictions for each* i*:*

Restriction I

$\Gamma_{\mathsf{gid}_i} \cup \Gamma'$ *must fail to satisfy* $\mathbb{S}$*, where* $\Gamma' := \{(t', \vec{x}_{A,t'}) \mid t' \in \mathcal{S}'\}$*.*

Restriction II

$\forall\delta$ *related with* $\Gamma_{\mathsf{gid}_i} \cup \Gamma'$*, when* $\mathbb{S}$ *accepts* δ*, i.e.,* $\vec{1} \in \mathsf{span}\langle M_\delta\rangle$ *s.t.* $M_\delta := (M_j)_{\gamma(j)=1}$*, there exists* j *s.t.* $\mathsf{FindNode}(\mathsf{gid}_i, (t, \vec{x}_{A,t}), \mathsf{rt}) \in r\ell_{\mathsf{v}_t} = \{\mathsf{FindNode}(\mathsf{rl}_t) \mid \mathsf{rl}_t \in RL_t\}$ *for any* rt *and* $\mathsf{usk}_{\mathsf{gid}_i,(t,\vec{x}_{A,t}),\mathsf{rt}}$ *which is given to* $\mathcal{A}$*,* $t \in \mathcal{S} \setminus \mathcal{S}'$*,* $\rho(j) = (t, \vec{x}_{A,t})$ *or* $\neg(t, \vec{x}_{A,t})$*.*

The adversary must also give the challenger the public keys and the revocation patches for any corrupt authorities whose attributes appear in the access structure. Given it, the challenger flips a random coin $b \xleftarrow{\mathsf{U}} \{0, 1\}$*, and sends the adversary* $\mathsf{ct}^{(b)}_{\mathbb{S},\{\mathsf{v}_t\}}$*(obtained by running* $\mathsf{PUpdate}$ *and* Enc*).*

Phase 2

The adversary is allowed to issue a polynomial number of queries, $(\mathsf{gid}, (t, \vec{x}_{A,t}))$*, to the challenger or oracle* $\mathsf{KeyGen}(\mathsf{gparam}, t, \mathsf{ask}_t, \mathsf{rsk}_t, \cdot, \cdot)$ *for private keys, user secret key* $\mathsf{usk}_{\mathsf{gid}_i,(t,\vec{x}_{A,t}),\mathsf{rt}}$ *subject to the same restriction as before. The adversary is also allowed to issue a polynomial number of queries,* $(\{(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})\}_{t\in\mathcal{S}\setminus\mathcal{S}'}, \mathsf{v}_t)$*, to the challenger or oracle* $\mathsf{PUpdate}(t, \mathsf{rpk}_t, \mathsf{rsk}_t, \{\mathsf{FindNode}(\cdot,\cdot,\cdot)\}, \cdot)$ *for revocation patch* $\mathsf{CP}_{\mathsf{v}_t}$ *subject to the same restriction as before.*

Guess

The adversary outputs a guess b' *of* b*.*

The advantage of an adversary $\mathcal{A}$ *in the above game is defined as* $\mathsf{Adv}^{\mathsf{R-DMA-FE,PHCA}}_{\mathcal{A}}(\lambda) := |\Pr[b' = b] - 1/2|$ *for any security parameter* λ*. An R-DMA-FE scheme is adaptively payload-hiding secure against chosen plaintext attacks and static corruption of authorities if all polynomial time adversaries have at most a negligible advantage in the above game.*

Remark: We show toy examples of adversary's key queries. Suppose that, there are three attribute authorities t_1, t_2 and t_3. t_1, t_2 and t_3 manage each attribute A_{t_1}, B_{t_2}, and C_{t_3} respectively. We would like to specify $(A_{t_1} \wedge B_{t_2}) \vee (A_{t_1} \wedge C_{t_3})$ as the challenge access structure.
Case 1. If the adversary gets the valid attribute key of A_{t_1}, the security game can be continued. The adversary follows the restriction I.
Case 2. If the adversary gets the valid attribute key of A_{t_1} and the revoked key of B_{t_2}, the security game can be continued. The adversary follows not the restriction I but the restriction II.
Case 3. If the adversary gets the valid attribute keys of A_{t_1} and C_{t_3} and the revoked key of B_{t_2}, the security game is aborted. The adversary's keys satisfy not $(A_{t_1} \wedge B_{t_2})$ but $(A_{t_1} \wedge C_{t_3})$. That is, the adversary does not follow the restriction I or II. The restriction II means that the adversary must have at least one revoked attribute key for each combination of attribute keys which satisfies the challenge access structure. If the adversary has the valid attribute key of A_{t_1} and the revoked keys of B_{t_2} and C_{t_3}, the security game can be continued. The adversary follows not the restriction I but the restriction II.

2.2 Construction

Our proposal is based on DMA-FE [18], so we follow the notations in [18].

We define function $\tilde{\rho} : \{1, \ldots, \ell\} \to \{1, \ldots, d\}$ 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 $\mathbb{S} := (M, \rho)$. In the proposed scheme, we assume that $\tilde{\rho}$ is injective for $\mathbb{S} := (M, \rho)$ with ciphertext $\mathsf{ct}_{\mathbb{S}, \{\mathsf{v}_t\}}$. In the description of the scheme, we assume that input vector $\vec{x}_{\mathsf{f},t} := (x_{\mathsf{f},t,1}, \ldots, x_{\mathsf{f},t,n_{\mathsf{f},t}})$ is normalized such that $x_{\mathsf{f},t,1} := 1$. (If $\vec{x}_{\mathsf{f},t}$ is not normalized, we can change it to a normalized one by $(1/x_{\mathsf{f},t,1}) \cdot \vec{x}_{\mathsf{f},t}$ assuming that $x_{\mathsf{f},t,1}$ is non-zero). In addition, we assume that input vector $\vec{v}_{R,\tilde{\rho}(i),\mathsf{v}_{\tilde{\rho}(i)},j} := (v_{R,\tilde{\rho}(i),\mathsf{v}_{\tilde{\rho}(i)},j,1}, \ldots, v_{R,\tilde{\rho}(i),\mathsf{v}_{\tilde{\rho}(i)},j,2\mathsf{h}_{\tilde{\rho}(i)}+4})$ satisfies that $v_{R,\tilde{\rho}(i),\mathsf{v}_{\tilde{\rho}(i)},j,2\mathsf{h}_{\tilde{\rho}(i)}+4} \neq 0$. We use the notations in [23] (which is the full version of this paper) for DPVS, e.g., $(x_1, \ldots, x_N)_{\mathbb{B}}$, $(y_1, \ldots, y_N)_{\mathbb{B}^*}$ for $x_i, y_i \in \mathbb{F}_q$, and $\vec{e}_{\mathsf{f},t,j}$. For matrix, $X := (\chi_{i,j})_{i,j=1,\ldots,N} \in \mathbb{F}_q^{N \times N}$ and element $\boldsymbol{v}$ in $N-$dimentional $\mathbb{V}$, $X(\boldsymbol{v})$ denotes $\sum_{i=1,j=1}^{N,N} \chi_{i,j} \phi_{i,j}(\boldsymbol{v})$ using canonical maps $\{\phi_{i,j}\}$. Similarly, for matrix $(\vartheta_{i,j}) := (X^{-1})^T$, $(X^{-1})^T(\boldsymbol{v}) := \sum_{i=1,j=1}^{N,N} \vartheta_{i,j} \phi_{i,j}(\boldsymbol{v})$. It holds that $e(X(\boldsymbol{x}), (X^{-1})^T(\boldsymbol{y})) = e(\boldsymbol{x}, \boldsymbol{y})$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$. In this paper, $\mathsf{f}(\in \{A, R\})$ is the subscript related to each functionality, that is, $\mathsf{f} = A$ is the subscript related to access control functionality and $\mathsf{f} = R$ is related to revocation functionality. The mapping $\psi_t : \{0,1\}^* \times \{(t, \vec{x}_{A,t})\} \times (\mathbb{N} \cup \{0\}) \to \{0,1\}^{\mathsf{h}_t}$ takes a user's global identifier gid, user's attribute $(t, \vec{x}_{A,t})$ and the number of return (after gid's $(t, \vec{x}_{A,t})$ revocation) rt, and outputs a user's label $L_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}$ assigned to the leaf node of a user binary tree managed by an attribute authority t, where h_t represents the height of the user binary tree. The one-to-one mapping $\Phi_t : \{*, 0, 1\} \times \{0, \ldots, \mathsf{h}_t\} \to \{3, \ldots, 2\mathsf{h}_t + 3\}$ takes 0, 1 (assigned to the edges of the user binary tree managed by an attribute authority t) or $*$ (assigned to the root node of it) and the depth of it, then outputs the positions of non-zero elements of the vector $\vec{v}_{R,\tilde{\rho}(i),\mathsf{v}_{\tilde{\rho}(i)},j}$. For example, $\Phi_t(*,0) := 3$, and, in general, $\Phi_t(a,b) := 2(b+1)+a$ where $a = 0,1$ and $b = 1,\ldots,\mathsf{h}_t$. We also defines L-list as the history of issuing user's key.

```
$\mathsf{GSetup}(1^\lambda)$ :
  $\mathsf{param}_{\mathbb{G}} := (q,\mathbb{G},\mathbb{G}_T,G,\boldsymbol{e}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda),\ H:\{0,1\}^* \to \mathbb{G};$
return $\mathsf{gparam} := (\mathsf{param}_{\mathbb{G}}, H)$.
  Remark: Given $\mathsf{gparam}$, the following values can be computed by anyone and
          shared by all parties: $G_0 := H(0^\lambda),\ G_1 := H(0^{\lambda-1} \parallel 1),\ g_T := \boldsymbol{e}(G_0,G_1)$.
$\mathsf{ASetup}(\mathsf{gparam}, t, n_{A,t}, N_{max,t}(=2^{\mathsf{h}_t}), \varphi_t)$ :
  $\mathsf{param}_{\mathbb{V}_{A,t}} := (q,\mathbb{V}_{A,t},\mathbb{G}_T,\mathbb{A}_{A,t},\boldsymbol{e}) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, 6n_{A,t}+1, \mathsf{param}_{\mathbb{G}})$,
  $X_{A,t} \xleftarrow{\mathsf{U}} GL(6n_{A,t}+1,\mathbb{F}_q),\ \boldsymbol{b}_{A,t,i} := X_{A,t}((0^{i-1},G_0,0^{6n_{A,t}+1-i}))$for $i = 1,\ldots,6n_{A,t}+1$,
  Set $\mathbb{B}_{A,t} := (\boldsymbol{b}_{A,t,i})_{i=1,\ldots,6n_{A,t}+1}$,
  $\hat{\mathbb{B}}_{A,t} := (\boldsymbol{b}_{A,t,1},\ldots,\boldsymbol{b}_{A,t,2n_{A,t}},\boldsymbol{b}_{A,t,5n_{A,t}+1},\ldots,\boldsymbol{b}_{A,t,6n_{A,t}+1})$,
  $\mathsf{ask}_t := X_{A,t},\ \mathsf{apk}_t := (\mathsf{param}_{\mathbb{V}_{A,t}}, \hat{\mathbb{B}}_{A,t})$,
  Run $\mathbf{CS.Setup}(N_{max,t})$ which takes the maximum number of users and outputs
  the user's binary tree. Then, assign a random value $\varsigma_{\nu_i} \in \mathbb{F}_q^\times$ to each leaf node $\nu_i$ in $\mathcal{BT}_t$.[7]
  $n_{R,t} := 4 + 2\log_2 N_{max,t} + \varphi_t$,
  $\mathsf{param}_{\mathbb{V}_{R,t}} := (q,\mathbb{V}_{R,t},\mathbb{G}_T,\mathbb{A}_{R,t},\boldsymbol{e}) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, 6n_{R,t}+1, \mathsf{param}_{\mathbb{G}})$,
  $X_{R,t} \xleftarrow{\mathsf{U}} GL(6n_{R,t}+1,\mathbb{F}_q),\ \boldsymbol{b}_{R,t,i} := X_{R,t}((0^{i-1},G_0,0^{6n_{R,t}+1-i}))$ for $i = 1,\ldots,6n_{R,t}+1$,
  Set $\mathbb{B}_{R,t} := (\boldsymbol{b}_{R,t,i})_{i=1,\ldots,6n_{R,t}+1}$,
  $\hat{\mathbb{B}}_{R,t} := (\boldsymbol{b}_{R,t,1},\boldsymbol{b}_{R,t,2\mathsf{h}_t+5},\ldots,\boldsymbol{b}_{R,t,n_{R,t}},\boldsymbol{b}_{R,t,n_{R,t}+1},\boldsymbol{b}_{R,t,n_{R,t}+2\mathsf{h}_t+5},\ldots,\boldsymbol{b}_{R,t,2n_{R,t}},$
          $\boldsymbol{b}_{R,t,5n_{R,t}+1},\boldsymbol{b}_{R,t,5n_{R,t}+2\mathsf{h}_t+5},\ldots,\boldsymbol{b}_{R,t,6n_{R,t}+1})$,
  $\mathsf{rsk}_t := (X_{R,t},\mathcal{BT}_t,\Phi_t,\psi_t),\ \mathsf{rpk}_t := (\mathsf{param}_{\mathbb{V}_{R,t}},\hat{\mathbb{B}}_{R,t},\varphi_t)$,
return $(\mathsf{ask}_t,\mathsf{apk}_t,\mathsf{rsk}_t,\mathsf{rpk}_t)$.
$\mathsf{PUpdate}(t,\mathsf{rpk}_t,\mathsf{rsk}_t,r\ell_{\mathsf{v}_t} := \{\mathsf{FindNode}(\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt})\},\mathsf{v}_t)$ :
  Run $\mathbf{CS.Cover}(\mathcal{BT}_t, r\ell_{\mathsf{v}_t})$ (which takes the user's binary tree and the revocation list)
  and outputs the covering set $CV_{r\ell_{\mathsf{v}_t}} = \{S_{i'_1},\ldots,S_{i'_{m'_{\mathsf{v}_t}}}\}$,
  for $j = 1,\ldots,m'_{\mathsf{v}_t}(\leq \varphi_t)$,
    $\eta^{[1]}_{t,\mathsf{v}_t,j},\eta^{[2]}_{t,\mathsf{v}_t,j},\eta^{[3]}_{t,\mathsf{v}_t,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
    $d_j, d_{j,a}, r_{t,\mathsf{v}_t,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$ s.t. $d_{j,0}+\ldots+d_{j,|ID(i'_j)|} = d_j$, for $a = 0,\ldots,|ID(i'_j)|$,
      for $1 \leqq z \leqq 2\mathsf{h}_t+4$,
        $v_{R,t,\mathsf{v}_t,j,z} = \begin{cases} d_j & (z=2) \\ -d_{j,a} & (z = \Phi_t(ID(i'_j)[a],a); 0 \leqq a \leqq |ID(i'_j)|) \\ r_{t,\mathsf{v}_t,j} & (z = 2\mathsf{h}_t+4) \\ 0 & (else) \end{cases}$
```

[7] $N_{max,t}$ is smaller than q for assigning ς_{ν_i} to each leaf node uniquely.

$$\boldsymbol{p}^{[1]}_{t,\mathsf{v}_t,j} = (\overbrace{\vec{v}_{R,t,\mathsf{v}_t,j}, 0^{\varphi_t}}^{n_{R,t}}, \overbrace{0^{n_{R,t}}}^{n_{R,t}}, \overbrace{0^{3n_{R,t}}}^{3n_{R,t}}, \overbrace{0^{n_{R,t}}}^{n_{R,t}}, \overbrace{\eta^{[1]}_{t,\mathsf{v}_t,j}}^{1})_{\mathbb{B}_{R,t}},$$

$$\boldsymbol{p}^{[2]}_{t,\mathsf{v}_t,j} = (\overbrace{0^{n_{R,t}}}^{n_{R,t}}, \overbrace{\vec{v}_{R,t,\mathsf{v}_t,j}, 0^{\varphi_t}}^{n_{R,t}}, \overbrace{0^{3n_{R,t}}}^{3n_{R,t}}, \overbrace{0^{n_{R,t}}}^{n_{R,t}}, \overbrace{\eta^{[2]}_{t,\mathsf{v}_t,j}}^{1})_{\mathbb{B}_{R,t}},$$

$$\boldsymbol{p}^{[3]}_{t,\mathsf{v}_t,j} = (\overbrace{0^{n_{R,t}}}^{n_{R,t}}, \overbrace{0^{n_{R,t}}}^{n_{R,t}}, \overbrace{0^{3n_{R,t}}}^{3n_{R,t}}, \overbrace{\vec{v}_{R,t,\mathsf{v}_t,j}, 0^{\varphi_t}}^{n_{R,t}}, \overbrace{\eta^{[3]}_{t,\mathsf{v}_t,j}}^{1})_{\mathbb{B}_{R,t}},$$

return $\mathsf{CP}_{\mathsf{v}_t} = (\mathsf{v}_t, \{\boldsymbol{p}^{[1]}_{t,\mathsf{v}_t,j}, \boldsymbol{p}^{[2]}_{t,\mathsf{v}_t,j}, \boldsymbol{p}^{[3]}_{t,\mathsf{v}_t,j}\}_{j=1}^{m'_{\mathsf{v}_t}})$.

$\mathsf{KeyGen}(\mathsf{gparam}, t, \mathsf{ask}_t, \mathsf{rsk}_t, \mathsf{gid}, \vec{x}_{A,t} := (x_{A,t,1}, \dots, x_{A,t,n_{A,t}}) \in \mathbb{F}_q^{n_{A,t}} \setminus \{\vec{0}\}$ s.t. $x_{A,t,1} := 1)$:

$G_{\mathsf{gid}} (= \delta G_1) := H(\mathsf{gid}) \in \mathbb{G}, \vec{\varphi}_{\mathsf{f},t} := (\varphi_{\mathsf{f},t,1}, \dots, \varphi_{\mathsf{f},t,n_{\mathsf{f},t}}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_{\mathsf{f},t}}$ for $\mathsf{f} = A, R$,

$\boldsymbol{b}^*_{\mathsf{f},t,i} := (X^{-1}_{\mathsf{f},t})^T((0^{i-1}, G_1, 0^{6n_{\mathsf{f},t}+1-i}))$ for $\mathsf{f} = A, R$,

Set $\mathbb{B}^*_{\mathsf{f},t} = (\boldsymbol{b}^*_{\mathsf{f},t,i})_{i=1,\dots,6n_{\mathsf{f},t}+1}$ for $\mathsf{f} = A, R$,

If $(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt}')$ exists in L-list, then set $\mathsf{rt} = \mathsf{rt}' + 1$ and change $(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt}')$ to $(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})$ in L-list,

If $(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt}')$ does not exist in L-list, then set $\mathsf{rt} = 0$ and add $(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})$ to L-list,

Assign $(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})$ to $\mathsf{FindNode}(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})$ − th leaf node of $\mathcal{BT}_t$,

$L_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}} = \psi_t(\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt})$,

Retrieve $\varsigma_{ID^{-1}(L_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}})}$ from $\mathcal{BT}_t$ and we define it as $\varsigma_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}$,

$$\boldsymbol{k}^*_{A,t} := (\overbrace{\vec{x}_{A,t}}^{n_{A,t}}, \overbrace{\delta\vec{x}_{A,t}}^{n_{A,t}}, \overbrace{0^{2n_{A,t}}}^{2n_{A,t}}, \overbrace{\vec{\varphi}_{A,t}}^{n_{A,t}}, \overbrace{\varsigma_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}\vec{x}_{A,t},}^{n_{A,t}} \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_{A,t}},$$

$\mathsf{uak}_{\mathsf{gid},(t,\vec{x}_{A,t})} := (\mathsf{gid}, (t, \vec{x}_{A,t}), \boldsymbol{k}^*_{A,t})$,

$\vec{x}_{R,t} := (x_{R,t,1}, \dots, x_{R,t,2\mathsf{h}_t+4})$,

for $z = 1, \dots, 2\mathsf{h}_t + 4 (= 4 + 2\log_2 N_{max,t})$,

$$x_{R,t,z} = \begin{cases} 1 & (z = 1) \\ \gamma & (z = 2, \Phi_t(L_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}[a], a); 0 \leqq a \leqq |L_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}| (= \mathsf{h}_t) \\ 0 & (else) \end{cases}$$

$$\boldsymbol{k}^*_{R,t} := (\overbrace{\vec{x}_{R,t}, 0^{\varphi_t}}^{n_{R,t}}, \overbrace{\delta\vec{x}_{R,t}, 0^{\varphi_t}}^{n_{R,t}}, \overbrace{0^{2n_{R,t}}}^{2n_{R,t}}, \overbrace{\vec{\varphi}_{R,t}}^{n_{R,t}} \overbrace{\varsigma_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}\vec{x}_{R,t}, 0^{\varphi_t}}^{n_{R,t}} \overbrace{\ 0\ }^{1})_{\mathbb{B}^*_{R,t}},$$

$\mathsf{uik}_{L_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}} := (\mathsf{gid}, (t, \vec{x}_{A,t}), \mathsf{rt}, \boldsymbol{k}^*_{R,t})$,

return $\mathsf{usk}_{L_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}} := (\mathsf{rt}, \mathsf{uak}_{\mathsf{gid},(t,\vec{x}_{A,t})}, \mathsf{uik}_{L_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}})$.

$\mathsf{Enc}(\{\mathsf{apk}_t, \mathsf{rpk}_t, \mathsf{CP}_{\mathsf{v}_t}\}, m, \mathbb{S})$:

$w_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$ for $i = 1, \dots, \ell$, $s'_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$,

$\vec{f}_A \xleftarrow{\mathsf{U}} \mathbb{F}_q^r,\ \vec{s}_A^{\,T} := (s_{A,1},\dots,s_{A,\ell})^T := M\cdot\vec{f}_A^{\,T},\ s_{A,0} := \vec{1}\cdot\vec{f}_A^{\,T},$

$\vec{f'}_A \xleftarrow{\mathsf{R}} \mathbb{F}_q^r$ s.t. $\vec{1}\cdot\vec{f'}_A^{\,T} = s'_0,\ \vec{s'}_A^{\,T} := (s'_{A,1},\dots,s'_{A,\ell})^T := M\cdot\vec{f'}_A^{\,T},$

$\vec{f}_R \xleftarrow{\mathsf{U}} \mathbb{F}_q^r,\ \vec{s}_R^{\,T} := (s_{R,1},\dots,s_{R,\ell})^T := M\cdot\vec{f}_R^{\,T},\ s_{R,0} := \vec{1}\cdot\vec{f}_{R,0}^{\,T},$

$\vec{f'}_R \xleftarrow{\mathsf{R}} \mathbb{F}_q^r$ s.t. $\vec{1}\cdot\vec{f'}_R^{\,T} = -s'_0,\ \vec{s'}_R^{\,T} := (s'_{R,1},\dots,s'_{R,\ell})^T := M\cdot\vec{f'}_R^{\,T},$

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

  $\eta_{A,i},\ \theta_{A,i},\ \theta'_{A,i},\ \theta''_{A,i} \xleftarrow{\mathsf{U}} \mathbb{F}_q,$

  if $\rho(i) = (t, \vec{v}_{A,i} := (v_{A,i,1},\dots,v_{A,i,n_{A,t}}) \in \mathbb{F}_q^{n_{A,t}} \setminus \{\vec{0}\}$ s.t. $v_{A,i,n_{A,t}} \neq 0),$

$$\begin{aligned}\boldsymbol{c}_{A,i} := (&\overbrace{s_{A,i}\vec{e}_{A,t,1} + \theta_{A,i}\vec{v}_{A,i}}^{n_{A,t}}, \overbrace{s'_{A,i}\vec{e}_{A,t,1} + \theta'_{A,i}\vec{v}_{A,i}}^{n_{A,t}}, \overbrace{0^{2n_{A,t}}}^{2n_{A,t}}, \overbrace{0^{n_{A,t}}}^{n_{A,t}},\\ &\overbrace{w_i\vec{e}_{A,t,1} + \theta''_{A,i}\vec{v}_{A,i}}^{n_{A,t}}, \overbrace{\eta_{A,i}}^{1})_{\mathbb{B}_{A,t}},\end{aligned}$$

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

$$\boldsymbol{c}_{A,i} := (\overbrace{s_{A,i}\vec{v}_{A,i}}^{n_{A,t}}, \overbrace{s'_{A,i}\vec{v}_{A,i}}^{n_{A,t}}, \overbrace{0^{2n_{A,t}}}^{2n_{A,t}}, \overbrace{0^{n_{A,t}}}^{n_{A,t}}, \overbrace{w_i\vec{v}_{A,i}}^{n_{A,t}}, \overbrace{\eta_{A,i}}^{1})_{\mathbb{B}_{A,t}},$$

  for $j = 1,\dots,m'_{\mathsf{v}_{\tilde{\rho}(i)}}$ (where $t = \tilde{\rho}(i)$),

   $\eta_{R,i,j},\ \theta_{R,i,j},\ \theta'_{R,i,j},\ \theta''_{R,i,j},\ \tau_{i,j},\ \tau'_{i,j},\ \tau''_{i,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q,$

$$\begin{aligned}\boldsymbol{c}_{R,i,j} = {}& s_{R,i}\boldsymbol{b}_{R,t,1} + \theta_{R,i,j}\boldsymbol{p}_t^{[1]}, \mathsf{v}_{t,j} + \tau_{i,j}\boldsymbol{b}_{R,t,j+2\mathsf{h}_t+4}\\ &+ s'_{R,i}\boldsymbol{b}_{R,t,n_{R,t}+1} + \theta'_{R,i,j}\boldsymbol{p}^{[2]}_{t,\mathsf{v}_t,j} + \tau'_{i,j}\boldsymbol{b}_{R,t,j+n_{R,t}+2\mathsf{h}_t+4}\\ &+ (-w_i)\boldsymbol{b}_{R,5n_{R,t}+1} + \theta''_{R,i,j}\boldsymbol{p}^{[3]}_{t,\mathsf{v}_t,j} + \tau''_{i,j}\boldsymbol{b}_{R,t,j+5n_{R,t}+2\mathsf{h}_t+4}\\ &+ \eta_{R,i,j}\boldsymbol{b}_{R,6n_{R,t}+1}\end{aligned}$$

   (Let $\eta'_{R,i,j} = \theta_{R,i,j}\eta^{[1]}_{t,\mathsf{v}_t,j} + \theta'_{R,i,j}\eta^{[2]}_{t,\mathsf{v}_t,j} + \theta''_{R,i,j}\eta^{[3]}_{t,\mathsf{v}_t,j} + \eta_{R,i,j}$)

   (We already defined $\vec{v}_{R,t,\mathsf{v}_t,j}$ in $\mathsf{PUpdate}$,)

$$\begin{aligned}= (&\overbrace{s_{R,i}\vec{e}_{R,t,1} + \theta_{R,i,j}\vec{v}_{R,t,\mathsf{v}_t,j}, 0^{j-1}, \tau_{i,j}, 0^{\varphi_t - j}}^{n_{R,t}},\\ &\overbrace{s'_{R,i}\vec{e}_{R,t,1} + \theta'_{R,i,j}\vec{v}_{R,t,\mathsf{v}_t,j}, 0^{j-1}, \tau'_{i,j}, 0^{\varphi_t - j}}^{n_{R,t}}, \overbrace{0^{2n_{R,t}}}^{2n_{R,t}}, \overbrace{0^{n_{R,t}}}^{n_{R,t}},\\ &\overbrace{-w_i\vec{e}_{R,t,1} + \theta''_{R,i,j}\vec{v}_{R,t,\mathsf{v}_t,j}, 0^{j-1}, \tau''_{i,j}, 0^{\varphi_t - j}}^{n_{R,t}}, \overbrace{\eta'_{R,i,j}}^{1})_{\mathbb{B}_{R,t}},\end{aligned}$$

$c_{d+1} := g_T^{s_{A,0}+s_{R,0}}m,\ \mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}} := (\mathbb{S}, \{\boldsymbol{c}_{A,i}, \{\boldsymbol{c}_{R,i,j}, \boldsymbol{p}^{[1]}_{t,\mathsf{v}_t,j}\}_{j=1}^{m'_{\mathsf{v}_t}}\}_{i=1}^{\ell}, c_{d+1})$[8]

return $\mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}}$.

[8] If an attribute authority t continues to make the past revocation patch available, $\mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}}$ does not have to include $\{\boldsymbol{p}^{[1]}_{t,\mathsf{v}_t,j}\}_{j=1}^{m'_{\mathsf{v}_t}}$. If $\mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}}$ includes it, an attribute authority t has only to publish the latest revocation patch.

$\mathsf{Dec}(\mathsf{gparam}, \{\mathsf{apk}_t, \mathsf{rpk}_t, \mathsf{usk}_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}\}, \mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}})$:

If $\mathbb{S} := (M, \rho)$ accepts $\Gamma := \{(t, \vec{x}_{A,t}) \in \mathsf{usk}_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}\}$,

then compute I and $\{\alpha_i\}_{i \in I}$ s.t. $\vec{1} = \sum_{i \in I} \alpha_i M_i$, where M_i is the $i-$th row of M, and

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

for each $i \in I$ (where $t = \tilde{\rho}(i)$),

pick $\{\boldsymbol{p}^{[1]}_{t,\mathsf{v}_t,j}\}_{j=1}^{m'_{\mathsf{v}_t}}$ from $\mathsf{ct}_{\mathbb{S},\{\mathsf{v}_t\}}$,

$K_i := \boldsymbol{e}(\boldsymbol{c}_{R,i,j}, \ \boldsymbol{k}^*_{R,t})$ for j s.t. $\boldsymbol{e}(\boldsymbol{p}^{[1]}_{t,\mathsf{v}_t,j}, \ \boldsymbol{k}^*_{R,t}) = 1$,

$$K := \prod_{i \in I \wedge \rho(i) = (t, \vec{v}_{A,i})} (\boldsymbol{e}(\boldsymbol{c}_{A,i}, \ \boldsymbol{k}^*_{A,t}) \cdot K_i)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i) = \neg(t, \vec{v}_{A,i})} (\boldsymbol{e}(\boldsymbol{c}_{A,i}, \ \boldsymbol{k}^*_{A,t})^{1/(\vec{v}_{A,i} \cdot \vec{x}_{A,t})} \cdot K_i)^{\alpha_i}$$

return $m' := c_{d+1}/K$.

[Correctness]

Here, the value g_T^s is written as $\mathsf{g_T}(s)$ in the way that the function e^x is written as $\exp(x)$.

$$K := \mathsf{g_T}(\sum_{i \in I \wedge \rho(i) = (t, \vec{v}_{A,i})} \alpha_i (s_{A,i} + \delta s'_{A,i} + w_i \varsigma_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}))$$

$$\mathsf{g_T}(\sum_{i \in I \wedge \rho(i) = \neg(t, \vec{v}_{A,i})} \alpha_i (\vec{v}_{A,i} \cdot \vec{x}_{A,t})^{-1} (s_{A,i} + \delta s'_{A,i} + w_i \varsigma_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}})(\vec{v}_{A,i} \cdot \vec{x}_{A,t}))$$

$$\mathsf{g_T}(\sum_{i \in I} \alpha_i (s_{R,i} + \delta s'_{R,i} - w_i \varsigma_{\mathsf{gid},(t,\vec{x}_{A,t}),\mathsf{rt}}))$$

$$= \mathsf{g_T}(\sum_{i \in I} (\alpha_i (s_{A,i} + s_{R,i}) + \delta \alpha_i (s'_{A,i} + s'_{R,i}))) = \mathsf{g_T}(s_{A,0} + s_{R,0}) = g_T^{s_{A,0} + s_{R,0}}$$

since $\sum_{i \in I} \alpha_i s_{\mathsf{f},i} = s_{\mathsf{f},0}$ for $\mathsf{f} = A, R$, $\sum_{i \in I} \alpha_i s'_{A,i} = s'_0$, $\sum_{i \in I} \alpha_i s'_{R,i} = -s'_0$.

Encoding of Attribute Vector for Revocation and Toy Example. The user is assigned a unique leaf node of the attribute binary tree used to manage users having the attribute. Then, the label is given to the user according to the path from the root node to the leaf node. At that time, the attribute vector is constructed according to the user's label and the covering node in the complete subtree method. The mapping Φ_t works to associate the basis in DPVS and the root node and the edges. The encoding of the attribute vector is as follows:

$$\vec{x}_{R,t} := (\overbrace{1}^{1}, \overbrace{\text{random value } \gamma}^{1}, \overbrace{\text{root node and each edge}(\gamma \text{ or } 0)}^{2\mathsf{h}_t+1}, \overbrace{0}^{1}),$$

$$\vec{v}_{R,t,\mathsf{v}_t,j} := (\overbrace{0}^{1}, \overbrace{\text{random value } d}^{1}, \overbrace{\text{root node and each edge}(-(d - \text{shared value})d_a \text{ or } 0)}^{2\mathsf{h}_t+1}, \\ \overbrace{\text{random value } r}^{1}),$$

We show a toy example in Fig. 1.

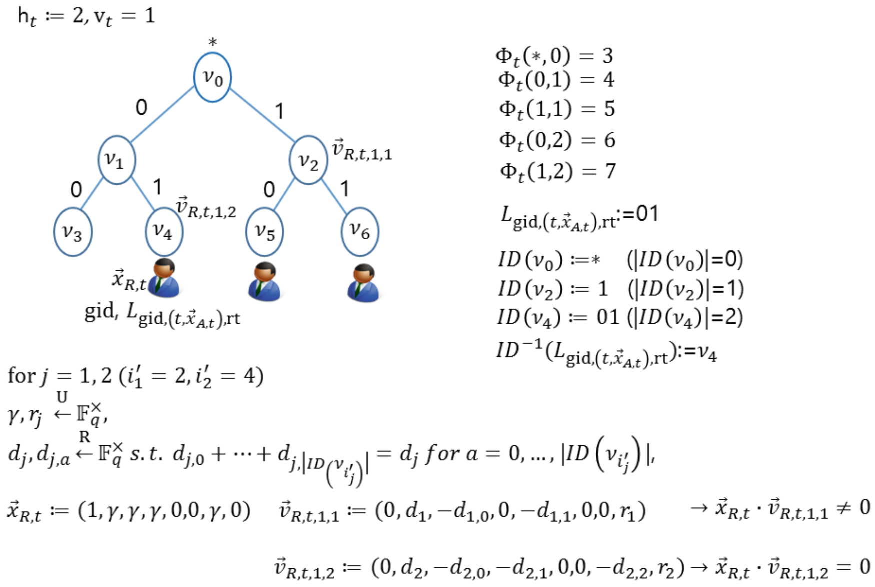
$\mathsf{h}_t := 2, \mathsf{v}_t = 1$
*
v_0
0
1
v_1
v_2
$\vec{v}_{R,t,1,1}$
0
1
0
1
$\vec{v}_{R,t,1,2}$
v_3
v_4
v_5
v_6
$\vec{x}_{R,t}$
gid, $L_{\mathrm{gid},(t,\vec{x}_{A,t}),\mathrm{rt}}$
$\Phi_t(*,0) = 3$
$\Phi_t(0,1) = 4$
$\Phi_t(1,1) = 5$
$\Phi_t(0,2) = 6$
$\Phi_t(1,2) = 7$
$L_{\mathrm{gid},(t,\vec{x}_{A,t}),\mathrm{rt}} := 01$
$ID(v_0) := *$ $(|ID(v_0)| = 0)$
$ID(v_2) := 1$ $(|ID(v_2)| = 1)$
$ID(v_4) := 01$ $(|ID(v_4)| = 2)$
$ID^{-1}(L_{\mathrm{gid},(t,\vec{x}_{A,t}),\mathrm{rt}}) := v_4$
for $j = 1, 2$ $(i'_1 = 2, i'_2 = 4)$
$\gamma, r_j \overset{\mathrm{U}}{\leftarrow} \mathbb{F}_q^{\times},$
$d_j, d_{j,a} \overset{\mathrm{R}}{\leftarrow} \mathbb{F}_q^{\times}$ $s.t.$ $d_{j,0} + \cdots + d_{j,|ID(v_{i'_j})|} = d_j$ for $a = 0, \ldots, |ID(v_{i'_j})|,$
$\vec{x}_{R,t} := (1, \gamma, \gamma, \gamma, 0, 0, \gamma, 0)$ $\vec{v}_{R,t,1,1} := (0, d_1, -d_{1,0}, 0, -d_{1,1}, 0, 0, r_1)$ $\rightarrow \vec{x}_{R,t} \cdot \vec{v}_{R,t,1,1} \neq 0$
$\vec{v}_{R,t,1,2} := (0, d_2, -d_{2,0}, -d_{2,1}, 0, 0, -d_{2,2}, r_2) \rightarrow \vec{x}_{R,t} \cdot \vec{v}_{R,t,1,2} = 0$

Fig. 1. Encoding of vector for revocation as a toy example

Comparison with the DMA-FE Scheme [18]. We show comparison with the DMA-FE scheme [18] in the full version of this paper [23].

2.3 Performance

We show a comparison of parameter size with previous works in Tables 3, 4 and 5. The construction of revocable-storage ABE supporting indirect revocation [11,21] is built in composite order bilinear groups. (The construction of [2,7,10,13,18] and our scheme is built in prime order bilinear groups.) Therefore, those are outside the scope of comparison. In Tables 3, 4 and 5, SK, PK, MSK, CT and UI represent the bit length of (user's) Secret Key, Public Key, Master Secret Key, Ciphertext and Update Information, respectively. UI is update key (in [2]) or revocation patch (in our scheme). $|\mathbb{G}|, |\mathbb{G}_T|, |\mathbb{Z}_q|$ and $|\mathbb{F}_q|$ represent the bit length of element in $\mathbb{G}, \mathbb{G}_T, \mathbb{Z}_q$ and $\mathbb{F}_q$, respectively. CA and AA represent Central Authority and Attribute Authority. Γ_{max} represents the maximum number of attributes in the system. Γ represents the number of attribute in user's secret keys or ciphertexts. ℓ is the size of rows in the LSSS matrix. h is the height of user's (binary) tree in the system. h_t is the height of user's (binary) tree managed by an attribute authority t. φ is the upper bound for the number of subsets in the cover in the system. φ_t is the upper bound for the number of subsets in the cover for an attribute category t. R means the number of revoked user in the system. R_t means the number of revoked user's attribute managed by an attribute authority t. n_t is the dimension of attribute vector in the category t. We define function

$\tilde{\rho} : \{1, \ldots, \ell\} \rightarrow \{1, \ldots, d\}$ 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 $\mathbb{S} := (M, \rho)$. We assume $\tilde{\rho}$ is injective for access structure with ciphertext. We also define $\hat{\rho} : \{1, \ldots, \Gamma\} \rightarrow \{1, \ldots, d\}$. Tables 1, 2, 3, 4 and 5 show that our proposal has more advantageous funtionalities in exchange for increasing the parameter size.

Table 3. Comparison of parameter size with previous works

Schemes	PK	MSK
AI09 [2]	$\lvert\mathbb{G}_T\rvert + (\Gamma_{max} + \varphi + 1)\lvert\mathbb{G}\rvert$	$2^{\mathsf{h}+1}\lvert\mathbb{Z}_q\rvert$
DDM15 [7]	$\lvert\mathbb{G}_T\rvert + 111\lvert\mathbb{G}\rvert$	$111\lvert\mathbb{G}\rvert$
H15 [10]	$2\lvert\mathbb{G}\rvert$ (CA) $\lvert\mathbb{G}_T\rvert + \lvert\mathbb{G}\rvert$ (AA)	$2\lvert\mathbb{Z}_q\rvert$ (CA) $2\lvert\mathbb{Z}_q\rvert$ (AA)
L12 [13]	$2\lvert\mathbb{G}_T\rvert + 48\lvert\mathbb{G}\rvert$ (AA)	$24\lvert\mathbb{G}\rvert + 48\lvert\mathbb{Z}_q\rvert$ (AA)
OT13 [18]	$(10n_t^2 + 7n_t + 1)\lvert\mathbb{G}\rvert$ (AA)	$(25n_t^2 + 10n_t + 1)\lvert\mathbb{F}_q\rvert$ (AA)
This work	$(18n_t^2 + 9n_t + 18\varphi_t^2 + 99\varphi_t$ $+36\mathsf{h}_t\varphi_t + 48\mathsf{h}_t + 101)\lvert\mathbb{G}\rvert$ (AA)	$(18n_t^2 + 9n_t + 14\mathsf{h}_t^2 + \varphi_t^2 + 6\mathsf{h}_t\varphi_t$ $+ 16\mathsf{h}_t + 8\varphi_t + 2^{\mathsf{h}_t} + 17)\lvert\mathbb{F}_q\rvert$ (AA)

Table 4. Comparison of parameter size with previous works (cont.)

Schemes	SK	CT
AI09 [2]	$2(\ell + 1)\log(2^{\mathsf{h}})\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert + (\Gamma + 1 + \mathsf{R}\log(2^{\mathsf{h}}/\mathsf{R}))\lvert\mathbb{G}\rvert$ (Direct) $\lvert\mathbb{G}_T\rvert + (\Gamma + 2)\lvert\mathbb{G}\rvert$ (Indirect)
DDM15 [7]	$(5 + 16\ell + 32\log^2(2^{\mathsf{h}}))\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert + (16\Gamma + 64\mathsf{R} - 27)\lvert\mathbb{G}\rvert$
H15 [10]	$(1 + \Gamma)\lvert\mathbb{G}\rvert$	$(1 + \ell)\lvert\mathbb{G}_T\rvert + 2(\ell + \mathsf{R})\lvert\mathbb{G}\rvert$
L12 [13]	$12\Gamma\lvert\mathbb{G}\rvert$	$(\ell + 1)\lvert\mathbb{G}_T\rvert + 12\ell\lvert\mathbb{G}\rvert$
OT13 [18]	$\sum_{i'=1}^{\Gamma} n_{\hat{\rho}(i')}\lvert\mathbb{F}_q\rvert + \sum_{i'=1}^{\Gamma}$ $(5n_{\hat{\rho}(i')} + 1)\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert + \sum_{i=1}^{\ell}(5n_{\tilde{\rho}(i)} + 1)\lvert\mathbb{G}\rvert$
This work	$\sum_{i'=1}^{\Gamma} n_{\hat{\rho}(i')}\lvert\mathbb{F}_q\rvert + \sum_{i'=1}^{\Gamma}(6n_{\hat{\rho}(i')}$ $+12\mathsf{h}_t + 6\varphi_t + 26)\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert + \sum_{i=1}^{\ell}(6n_{\tilde{\rho}(i)} + 1 + \mathsf{R}_{\tilde{\rho}(i)}\log(2^{\mathsf{h}_{\tilde{\rho}(i)}}/$ $\mathsf{R}_{\tilde{\rho}(i)})(24\mathsf{h}_{\tilde{\rho}(i)} + 12\varphi_{\tilde{\rho}(i)} + 50))\lvert\mathbb{G}\rvert$

Table 5. Comparison of parameter size with previous works (cont.)

Schemes	UI
AI09 [2]	$2(\mathsf{R}\log(2^{\mathsf{h}}/\mathsf{R}))\lvert\mathbb{G}\rvert$
DDM15 [7]	-
H15 [10]	-
L12 [13]	-
OT13 [18]	-
This work	$(36\mathsf{h}_t\mathsf{R}_t\log(2^{\mathsf{h}_t}/\mathsf{R}_t) + 18\varphi_t\mathsf{R}_t\log(2^{\mathsf{h}_t}/\mathsf{R}_t) + 75\mathsf{R}_t\log(2^{\mathsf{R}_t}/\mathsf{R}_t))\lvert\mathbb{G}\rvert$ (AA)

2.4 Security of the Proposed R-DMA-FE

The DLIN assumption is given in the full version of this paper [23].

Theorem 1. *The proposed R-DMA-FE scheme is adaptively payload-hiding against chosen plaintext attacks and static corruption of authorities under the DLIN assumption in the random oracle model.*

The proof of Theorem 1 is given in the full version of this paper [23].

3 Conclusion

In this paper, we proposed the first DMA-FE scheme supporting patch revocation in which an encryptor does not have to care about the revocation list. Our proposed scheme realizes the revocation on the category level that is a generalization of attribute level for the first time. We proved that our construction is adaptively secure against chosen plaintext attacks and static corruption of authorities based on the DLIN assumption. (We note that DMA-FE [18] of Okamoto and Takashima does not achieve the security against static corruption of authorities.) In the future, we will try to apply new techniques called indexing and consistent randomness amplification [17] to reduce the size of public parameters.

Acknowledgements. This work was supported in part by JSPS KAKENHI Grant Number 26330151 and JSPS and DST under the Japan - India Science Cooperative Program. The authors would like to thank anonymous reviewers for their useful comments.

References

1. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute-based encryption. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 248–265. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03298-1_16
2. Attrapadung, N., Imai, H.: Attribute-based encryption supporting direct/indirect revocation modes. In: Parker, M.G. (ed.) IMACC 2009. LNCS, vol. 5921, pp. 278–300. Springer, Heidelberg (2009). doi:10.1007/978-3-642-10868-6_17
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
4. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: ACM CCS 2008, pp. 417–426 (2008)
5. Boneh, D., Sahai, A., Waters, B.: Functional encryption: definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19571-6_16
6. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007). doi:10.1007/978-3-540-70936-7_28
7. Datta, P., Dutta, R., Mukhopadhyay, S.: Adaptively secure unrestricted attribute-based encryption with subset difference revocation in bilinear groups of prime order. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2016. LNCS, vol. 9646, pp. 325–345. Springer, Heidelberg (2016). doi:10.1007/978-3-319-31517-1_17

8. González-Nieto, J.M., Manulis, M., Sun, D.: Fully private revocable predicate encryption. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 350–363. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31448-3_26
9. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM CCS 2006, pp. 89–98 (2006)
10. Horváth, M.: Attribute-based encryption optimized for cloud computing. In: Italiano, G.F., Margaria-Steffen, T., Pokorný, J., Quisquater, J.-J., Wattenhofer, R. (eds.) SOFSEM 2015. LNCS, vol. 8939, pp. 566–577. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46078-8_47
11. Lee, K., Choi, S.G., Lee, D.H., Park, J.H., Yung, M.: Self-updatable encryption: time constrained access control with hidden attributes and better efficiency. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 235–254. Springer, Heidelberg (2013). doi:10.1007/978-3-642-42033-7_13
12. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011). doi:10.1007/978-3-642-20465-4_31
13. Lewko, A.B.: Functional encryption: new proof techniques and advancing capabilities. Ph.D. thesis, The University of Texas (2012)
14. Müller, S., Katzenbeisser, S., Eckert, C.: Distributed attribute-based encryption. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 20–36. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00730-9_2
15. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_3
16. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14623-7_11
17. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 349–366. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34961-4_22
18. Okamoto, T., Takashima, K.: Decentralized attribute-based signatures. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 125–142. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36362-7_9
19. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM CCS 2007, pp. 195–203 (2007)
20. Qian, J., Dong, X.: Fully secure revocable attribute-based encryption. J. Shanghai Jiaotong Univ. (Sci.) **16**(4), 490–496 (2011)
21. Sahai, A., Seyalioglu, H., Waters, B.: Dynamic credentials and ciphertext delegation for attribute-based encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 199–217. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32009-5_13
22. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). doi:10.1007/11426639_27
23. The full version of this paper. It will appear in the IACR Cryptology ePrint Archive. https://eprint.iacr.org/