Information Sciences 295 (2015) 221–231

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation

Yanfeng Shi [a,*], Qingji Zheng [b], Jiqiang Liu [a], Zhen Han [a]

[a] School of Computer and Information Technology, Beijing Jiaotong University, China
[b] Department of Computer Science, University of Texas at San Antonio, USA

ARTICLE INFO

Article history:
Received 5 September 2013
Received in revised form 5 May 2014
Accepted 12 October 2014
Available online 18 October 2014

Keywords:
Attribute-based encryption
Verifiable ciphertext delegation
Direct revocation

ABSTRACT

Attribute-based encryption (ABE) enables an access control mechanism by specifying access control policies among decryption keys and ciphertexts. In this paper, we propose a novel ABE variant, dubbed *directly revocable key-policy* ABE *with verifiable ciphertext delegation* (drvuKPABE), which supports direct revocation and verifiable ciphertext delegation. The drvuKPABE offers the following features which are promising in the data sharing applications: (1) it allows the trusted authority to revoke users by solely updating the revocation list while mitigating the interaction with non-revoked users, which is unlikely to indirectly revokable ABE; (2) it allows the third party to update ciphertexts with public information so that those non-revoked users cannot decrypt them; and (3) it enables any auditor (authorized by data owners) to verify whether the untrusted third party updated ciphertexts correctly or not. We formalize the syntax and security properties for drvuKPABE, and propose the construction based on the multilinear maps. Our solution attains the security properties under the $(d+3)$-Multilinear Decisional Diffie–Hellman assumption in the random oracle model.

1. Introduction

Cloud computing allows data users to outsource/share their data while enjoying affordable price and high scalability. Despite numerous advantages, data outsourcing hinders data owners managing outsourced data: how to preserve the privacy of outsourced data and enforce access control policies on accessing it. Fortunately, attribute-based encryption (ABE) can be the right cryptographic tool solving these concerns: Data owners can specify access control policy on outsourced data while encrypting it, and users can decrypt ciphertexts only if their attributes satisfy the access control policy.

However, pure ABE is not sufficient for data sharing applications since users' access rights are not static: a user's access right might be revoked if he/she leaves the organization. The variant, revocable ABE, can be a good candidate to fulfill this requirement, while still raising two related challenges: (i) how to mitigate the interaction with non-revoked users when the trusted authority revokes users? and (ii) how to forbid revoked users decrypting ciphertexts that were generated previously? To the best of our knowledge, revocable ABE in the literature cannot resolve these challenges simultaneously (The detail will be discussed in Section 1.1).

* Corresponding author at: School of Computer and Information Technology, Beijing Jiaotong University, 100044 Beijing, China. Tel.: +86 13488787156.
E-mail address: schwannrobben@gmail.com (Y. Shi).

http://dx.doi.org/10.1016/j.ins.2014.10.020
0020-0255/

Our contribution. We propose a novel cryptographic solution, dubbed *directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation* (drvuKPABE). The solution allows data owner to enforce access control policy on outsourced encrypted data while offering the promising properties as follows: (1) direct revocation: when revoking users, the trusted authority only needs to update the revocation list, without any interaction with non-revoked users; (2) ciphertext delegation: in order to assure that prior ciphertexts cannot be decrypted by revoked users, outsourced encrypted data can be updated by an untrusted third party (e.g., storage provider) with public known information. In addition, since the third party might be untrusted, it naturally leads to the third promising properties: (3) update verifiability: it allows any auditor (e.g., authorized by data owners) to verify whether ciphertexts were updated correctly from some prior revocation list to the current revocation list or not.

We formally define the syntax and security properties of drvuKPABE and present a provably secure construction satisfying them. Building on top of multilinear maps, our construction is proved secure under the $(d+3)$-MDDH assumption (see Section 2) in the random oracle model. For fairness, we summarize the properties of revocable KPABE schemes in Table 1.

1.1. Related work

Attributed-based encryption (ABE), first introduced in [23], is an effective cryptographic primitive to preserve data secrecy and enforce fine-grained access control policy simultaneously. There are two flavors of ABE according to how the access control policy is enforced: Key Policy ABE (KPABE) schemes (e.g., [12]) enforce access control policies in the decryption keys while Ciphertext Policy ABE (CPABE) schemes (e.g., [24,11,4,9,3]) enforce access control policies in the ciphertexts. Many variants of ABE have been proposed to providing promising properties, e.g., attribute-based keyword search [27], multi-authority attribute-based encryption [7,15] and attribute-based encryption with outsourcing decryption [14,13]. Among these features, how to construct revocable ABE is an active research topic.

Existing solutions for revocable ABE can be classified into two categories [1] depending on how to integrate revocation information: directly revocable ABE (e.g., [20,11,2]) and indirectly revocable ABE (e.g., [21,25]). In the setting of directly revocable ABE, the trusted authority makes the revocation list (i.e., listing all revoked users' identities) public known so that users can integrate revocation information into the ciphertext while encrypting data, by which any revoked user cannot decrypt ciphertexts even his/her owned credentials satisfy access control policies specified by ciphertexts. The advantage of this approach is that there is no need for the trusted authority to update decryption keys owned by non-revoked users. In the setting of indirectly revocable ABE, instead of publishing the revocation list, the trusted authority needs to communicate with non-revoked users and distribute new decryption keys to them. Despite that there is no need for users to know the revocation list when conducting encryption, this approach requires non-revoked users updating their decryption keys, which is undesirable in data sharing applications involving a large number of participants. Note that the mechanism in both approaches only assures that revoked users cannot decrypt ciphertexts generated after revocation. To prevent revoking users from decrypting ciphertexts that can be decrypted before revocation, some other mechanisms, e.g. proxy re-encryption [17,19,26] and sharing some secret key in advance [16], are introduced, which need the interaction between the proxy and the trusted authority (or data owners). Note that another work [22] considers the problem of updating ciphertexts in the setting of indirectly revocable ABE, where the third party (e.g., storage provider) can update stored ciphertexts without any interaction with either data owners or the trusted authority as long as the revocation event happens.

Different from prior solutions, we consider verifiable ciphertext delegation underlying the setting of directly revocable ABE, which enjoys the advantage that the trusted authority has no need to update decryption keys of non-revoked users while still being able to delegate the third party to update ciphertexts.

2. Preliminaries and notations

Let $x \xleftarrow{R} X$ denote selecting element x from the set X uniformly at random, and $\mathsf{UAtt} = \{\mathsf{at}_1, \ldots, \mathsf{at}_N\}$ be the attribute universe. Table 2 summaries the notations that are used through the paper.

Multilinear maps. The concept of multilinear maps was introduced in [6] and came to reality by the works [10,8]. Given security parameter ℓ and an ℓ-bit prime p, a $d+3$ multilinear map consists of $d+3$ groups $(G_0, G_1, \ldots, G_{d+2})$ with order p, and $d+2$ mappings $e_i : G_0 \times G_i \to G_{i+1}, i = 0, \ldots, d+1$. The multilinear maps should satisfy the following properties with respect to $i, i = 0, \ldots, d+1$: (i) given that $g_0 \in G_0$ is a generator of G_0, then $g_{i+1} = e_i(g_0, g_i)$ is a generator of G_{i+1}; (ii) $\forall \alpha, \beta \in \mathbb{Z}_p$, $e_i(g_0^{\alpha}, g_i^{\beta}) = e_i(g_0, g_i)^{\alpha\beta}$; and (iii) e_i can be efficiently computed.

$(d+3)$-Multilinear Decisional Diffie–Hellman assumption ($(d+3)$-MDDH). Given the multilinear maps and $g_0, g_0^{z_0}, \ldots, g_0^{z_d}, g_0^a, g_0^b, g_0^c, Z$, where $z_0, \ldots, z_d, a, b, c \leftarrow \mathbb{Z}_p$ and $Z \leftarrow G_{d+2}$, there exists no probabilistic polynomial algorithm $\mathcal{A}$ that can determine $g_{d+2}^{z_0 \cdots z_d abc} \stackrel{?}{=} Z$ with a non-negligible advantage with respect to security parameter ℓ, where the advantage is defined as

$$\left| \Pr\left[\mathcal{A}\left(g_{d+2}^{z_0 \cdots z_d abc}, g_0, g_0^{z_0}, \ldots, g_0^{z_d}, g_0^a, g_0^b, g_0^c\right) = 1\right] - \Pr\left[\mathcal{A}\left(Z, g_0, g_0^{z_0}, \ldots, g_0^{z_d}, g_0^a, g_0^b, g_0^c\right) = 1\right] \right|.$$

[1] There is another work [1] (i.e., hybrid approach) that allows data users/authority to choose the revocation types (i.e., direct or indirect revocation) on the fly. However, the revocation manner cannot be changed after being chosen.

Table 1
Property summary for revocable KPABE schemes in the literature and the solution in this paper. Direct revocation means that the trusted authority can solely update revocation list and there is no need to update non-revoked users' decryption key. Ciphertext delegation means that ciphertexts can be updated by the third party correspondingly when the revocation list is updated. Update verifiability means that the process of the third party updating ciphertexts can be accountable.

Scheme	Direct revocation	Ciphertext delegation	Update verifiability	Security assumption
Scheme 1 [2]	√	×	×	n-BDHE
Scheme 2 [2]	√	×	×	r-MEBDH
[1]	√	×	×	DBDH
[5]	×	×	×	DBDH
[22]	×	√	×	Three static assumptions
Our solution	√	√	√	$(d+3)$-MDDH

Table 2
Notations.

Notation	Description
ℓ, p	ℓ is a security parameter and p is an ℓ-bit prime
$G_0, \ldots, G_{d+2}$	Cyclic groups of order p
$e_i : G_0 \times G_i \to G_{i+1}$	Bilinear mapping from G_0 and G_i to G_{i+1}
$\mathsf{LSSS}(\mathsf{M}, \pi)$	A linear secret sharing scheme that is specified by matrix M and injective function π
P	An access control policy
R	Revocation list
max	The maximum number of attributes that are associated to a ciphertext

Linear secret sharing scheme. A linear secret sharing scheme (LSSS) can be used to represent an access control policy P via (M, π), where $\mathsf{M} = (\mathbb{Z}_p)^{l \times k}$ is an $l \times k$ matrix with entries belonging to $\mathbb{Z}_p$ and $\pi : \{1, \ldots, l\} \to \mathsf{UAtt}$ is an injective function that maps a row into an attribute. Given an attribute set $\mathsf{S} \subset \mathsf{UAtt}$, denote by $F(\mathsf{S}, \mathsf{P}) = 1$ if S satisfies access control policy P. A LSSS consists of two algorithms:

$\mathsf{Share}((\mathsf{M}, \pi), s)$: this algorithm is to distribute secret value s to attributes specified by π as follows: by selecting $v_2, \ldots, v_k \overset{R}{\leftarrow} \mathbb{Z}_p$, setting $\mathbf{v} = (s, v_2, \ldots, v_k)$ and computing $\lambda_{\pi(i)} = \mathsf{M}_i \cdot \mathbf{v}$ where M_i is the ith row of M, it assigns secret share $\lambda_{\pi(i)}$ to the attribute $\pi(i)$.

$\mathsf{Combine}(\mathsf{S}, (\lambda_{\pi(i)}, \ldots, \lambda_{\pi(l)}), (\mathsf{M}, \pi))$: this algorithm is to assemble the secret value from secret shares associated with the attributes as follows: selecting subset $I = \{i | \pi(i) \in \mathsf{S}\}$ such that the attribute set $\{\pi(i) | i \in I\}$ satisfies access control policy (M, π), and computing coefficients $c_i, i \in I$ such that $\sum_{i \in I} c_i \mathsf{M}_i = (1, 0, \ldots, 0)$, it sets the recovered secret to be $\sum_{i \in I} c_i \lambda_{\pi(i)} = s$.

The correctness of algorithm Combine is assured by the following lemma:

Lemma 1 [24]. *Let* (M, π) *be a* LSSS *representing an access control policy* P. *For all attributes in* S *that do not satisfy* P, *there is a polynomial-time algorithm that outputs vector* $\boldsymbol{w} = (w_1, \ldots, w_k) \in \mathbb{Z}_p^k$ *such that* $w_1 = 1$ *and* $\mathsf{M}_i \cdot \boldsymbol{w} = 0$ *for all* $i \in [1, \ldots, l]$, *where* $\pi(i) \in \mathsf{S}$.

Subset cover. Let T be a full binary tree with n leaves (representing n users), where each nodes are labeled, $\mathsf{depth}(x)$ denote the depth (i.e., number of hops from the root) of node x such that $\mathsf{depth}(root) = 0$. Let $\mathsf{path}(x) = \{x_{i_0} = root, \ldots, x_{i_{\mathsf{depth}(x)}} = x\}$ denote the path from the root to node x.

Given the full binary tree T, the subset cover technique [18] can be used to encode revoked users. Specifically, given a set of leaf nodes R (corresponding to a set of revoked users), $\mathsf{cover}(\mathsf{R})$ is defined as follows: $\forall x \in \mathsf{R}$, mark all nodes of $\mathsf{path}(x)$ and then $\mathsf{cover}(\mathsf{R})$ is the set of unmarked nodes that are the direct children of marked nodes in T. Consider the example in Fig. 1 where T contains 8 leaves $x_1, \ldots, x_8$. Given $\mathsf{R} = \{x_1, x_4\}$, by marking nodes in $\mathsf{path}(x_1) = \{x_{15}, x_{13}, x_9, x_1\}$ and $\mathsf{path}(x_4) = \{x_{15}, x_{13}, x_{10}, x_4\}$, we can see that $\mathsf{cover}(\mathsf{R}) = \{x_2, x_3, x_{14}\}$. The leaves covered by the nodes in $\mathsf{cover}(\mathsf{R})$ are non-revoked users.

3. drvuKPABE: syntax and security

Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation (drvuKPABE) is a natural extension for key-policy ABE where decryption keys are associated to access control policies and ciphertexts are associated to attributes, while offering two promising properties when the user revocation happens: (i) there is no need to update non-revoked users' decryption keys; and (ii) an untrusted third party is allowed to update ciphertexts with public information (without leaking any information of plaintext with respect to the ciphertext) and (iii) any auditor can verify whether the third party has executed the ciphertexts update honestly or not.

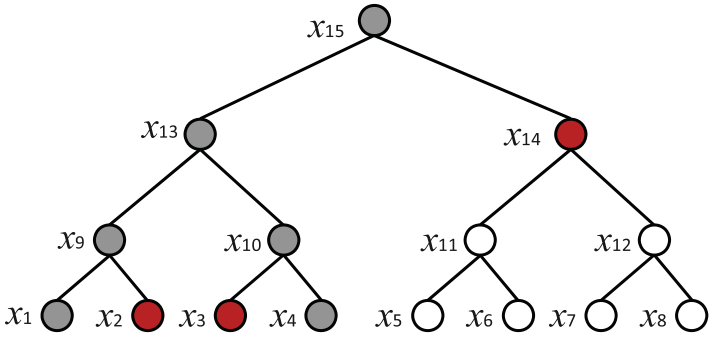

Fig. 1. Subset cover technique to encode the revocation list. Given the revocation list $\mathsf{R} = \{x_1, x_4\}$, the nodes of $\mathsf{path}(x_1)$ and $\mathsf{path}(x_4)$ are marked (in gray color), and then $\mathsf{cover}(\mathsf{R}) = \{x_2, x_3, x_{14}\}$ (in red color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Definition 1. A drvuKPABE scheme consists of seven algorithms as follows:

$(\mathsf{pm}, \mathsf{mk}) \leftarrow \mathsf{Setup}(1^\ell)$: this algorithm is run by the trusted authority, who is responsible for managing users and issuing decryption keys, to generate the master secret key mk and the public parameter pm. For brevity, we assume that the following algorithms implicitly take pm as part of inputs.

$\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mk}, \mathsf{P}, \mathsf{uid})$: given the access control policy P, this algorithm is run by the trusted authority to generate the decryption key sk for the user identified by uid.

$\mathsf{cph} \leftarrow \mathsf{Enc}(\mathsf{m}, \mathsf{S}, \mathsf{R})$: this algorithm is to encrypt the message m and output a ciphertext cph, where S is an attribute set and R is the revocation list.

$\{\mathsf{m}, \perp\} \leftarrow \mathsf{Dec}(\mathsf{cph}, \mathsf{sk})$: this algorithm is to decrypt the ciphertext cph with the decryption key sk. The decryption can be done successfully only if (i) the attribute set S associated to cph satisfies the access control policy P specified by the decryption key; and (ii) the user identity specified by the decryption key has not been revoked according to the revocation list specified by cph. Otherwise, it outputs an error message $\perp$.

$\mathsf{cph}' \leftarrow \mathsf{Update}(\mathsf{cph}, \mathsf{R}')$: this algorithm allows any third party to update ciphertext cph with respect to R to a new ciphertext cph′ with respect to a new revocation list R′, where $\mathsf{R} \subset \mathsf{R}'$.

$\{0, 1\} \leftarrow \mathsf{Verify}(\mathsf{cph}, \mathsf{cph}')$: this algorithm is run by any party (i.e., the data owner outsourcing cph) to verify whether cph′ is updated correctly from cph when the revocation list R (specified by cph) is changed to R′ (specified by cph'). It outputs 1 if the update is correct, and 0 otherwise.

The correctness of a drvuKPABE scheme can be defined as follows: given $(\mathsf{pm}, \mathsf{mk}) \leftarrow \mathsf{Setup}(1^\ell)$, for any message m, any attribute set S, and revocation lists R and R′ such that $\mathsf{R} \subset \mathsf{R}'$, let $\mathsf{cph} \leftarrow \mathsf{Enc}(\mathsf{m}, \mathsf{S}, \mathsf{R})$ and $\mathsf{cph}' \leftarrow \mathsf{Update}(\mathsf{cph}, \mathsf{R}')$, the drvuKPABE scheme is correct only if the following always holds: (i) $1 \leftarrow \mathsf{Verify}(\mathsf{cph}, \mathsf{cph}')$; (ii) given $\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mk}, \mathsf{P}, \mathsf{uid})$ where $F(\mathsf{S}, \mathsf{P}) = 1$ and $\mathsf{uid} \notin \mathsf{R}, \mathsf{Dec}(\mathsf{cph}, \mathsf{sk}) = \mathsf{m}$; and (iii) given $\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mk}, \mathsf{P}, \mathsf{uid})$ where $F(\mathsf{S}, \mathsf{P}) = 1$ and $\mathsf{uid} \notin \mathsf{R}'$, then $\mathsf{Dec}(\mathsf{cph}', \mathsf{sk}) = \mathsf{m}$.

The adversary model against drvuKPABE is the following: unauthorized data users (i.e., attributes do not satisfy the access control policy) and revoked data users (i.e., identities are shown in the revocation list) are malicious and try their best to learn any information from ciphertexts. The party (i.e., cloud) that transforms ciphertexts when the revocation list was updated might be malicious in the sense that it can perform the ciphertext update dishonestly. Specifically, given a probabilistic polynomial time adversary $\mathcal{A}$, a drvuKPABE is secure only if the following holds:

(i) Selective security against chosen-plaintext attack on original ciphertext. The adversary $\mathcal{A}$ (modeling unauthorized/revoked data users) cannot infer any information about the plaintext of an original ciphertext (i.e., without being updated) in the selective security model. This property is formalized by the selective security game on original ciphertext.

(ii) Selective security against chosen-plaintext attack on updated ciphertext. The adversary $\mathcal{A}$ (modeling revoked data users) cannot infer any information about the plaintext of an updated ciphertext in the selective security model. This property is formalized by the selective security game on updated ciphertext.

(iii) Update verifiability. The adversary $\mathcal{A}$ (modeling any third party that updates ciphertexts) cannot update ciphertexts incorrectly without being caught. This property is formalized by the verifiability game.

3.1. Selective security game on original ciphertext

Setup: the adversary $\mathcal{A}$ chooses an attribute set S^* and a revocation list R^*, and sends them to the challenger. The challenger runs Setup to produce pm, mk, sends pm to $\mathcal{A}$ and keeps mk secret.

Phase 1: $\mathcal{A}$ is allowed to query the following oracle in polynomially many times:

- $\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{uid}, \mathsf{P})$: If $F(\mathsf{S}^*, \mathsf{P}) = 1$ and $\mathsf{uid} \notin \mathsf{R}^*$ simultaneously, then abort. Otherwise, the challenger runs $\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mk}, \mathsf{P}, \mathsf{uid})$ and returns sk to $\mathcal{A}$.

Challenge: $\mathcal{A}$ chooses two message m_0 and m_1 of equal length, and sends them to the challenger. The challenger selects $\sigma \xleftarrow{R} \{0,1\}$, runs $\mathsf{cph}^* = \mathsf{Enc}(\mathsf{m}_\sigma, \mathsf{S}^*, \mathsf{R}^*)$ and forwards cph^* to $\mathcal{A}$.
Phase 2: $\mathcal{A}$ queries the oracle the same as in Phase 1.
Guess: $\mathcal{A}$ outputs a guess σ'. We say $\mathcal{A}$ wins this game if $\sigma = \sigma'$.

Definition 2. A drvuKPABE scheme achieves selective security on original ciphertext if the advantage of any adversary $\mathcal{A}$ winning the selective security on original ciphertext is negligible at most with respect to security parameter ℓ, where the advantage is defined as $\Pr[\sigma' = \sigma] - 1/2$.

3.2. Selective security game on updated ciphertext

Setup: the adversary $\mathcal{A}$ chooses an attribute set S^* and *two* revocation lists R and R^* where $\mathsf{R} \subset \mathsf{R}^*$, and sends them to the challenger. The challenger runs Setup to produce pm, mk, sends pm to $\mathcal{A}$ and keeps mk secret.
Phase 1: $\mathcal{A}$ is allowed to query the following oracle in polynomially many times:

- $\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{uid}, \mathsf{P})$: If $F(\mathsf{S}^*, \mathsf{P}) = 1$ and $\mathsf{uid} \notin \mathsf{R}^*$ simultaneously, then abort. Otherwise, the challenger runs $\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mk}, \mathsf{P}, \mathsf{uid})$ and returns sk to $\mathcal{A}$.

Challenge: $\mathcal{A}$ selects two message m_0 and m_1 of equal length, and sends them to the challenger. The challenger selects $\sigma \xleftarrow{R} \{0,1\}$, runs $\mathsf{cph} = \mathsf{Enc}(\mathsf{m}_\sigma, \mathsf{S}^*, \mathsf{R})$ and $\mathsf{cph}^* \leftarrow \mathsf{Update}(\mathsf{cph}, \mathsf{R}^*)$, and forwards cph^* to $\mathcal{A}$.
Phase 2: $\mathcal{A}$ queries the oracle the same as in Phase 1.
Guess: $\mathcal{A}$ outputs a guess σ'. We say $\mathcal{A}$ wins this game if $\sigma = \sigma'$.

Definition 3. A drvuKPABE scheme achieves selective security on updated ciphertext if the advantage of any adversary $\mathcal{A}$ winning the selective security on updated ciphertext is negligible at most with respect to security parameter ℓ, where the advantage is defined as $\Pr[\sigma' = \sigma] - 1/2$.

3.3. Verifiability game

Setup: the adversary $\mathcal{A}$ chooses an attribute set S^* and sends it to the challenger. The challenger runs Setup to produce pm, mk and sends pm to $\mathcal{A}$ and keeps mk secret.
Phase 1: $\mathcal{A}$ is allowed to query the following oracles in polynomially times:

- $\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{uid}, \mathsf{P})$: the challenger runs $\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mk}, \mathsf{P}, \mathsf{uid})$ and returns sk to $\mathcal{A}$.

Challenge: $\mathcal{A}$ selects a message m, and two revocation lists R and R^*, where $\mathsf{R} \subset \mathsf{R}^*$, and sends them to the challenger. The challenger runs $\mathsf{cph} = \mathsf{Enc}(\mathsf{m}, \mathsf{S}^*, \mathsf{R})$ and sends cph to $\mathcal{A}$.
Guess: $\mathcal{A}$ returns a updated ciphertext cph^* to the challenger with respect to the revocation list R^*. We say that $\mathcal{A}$ wins this game if $1 \leftarrow \mathsf{Verify}(\mathsf{cph}, \mathsf{cph}^*)$ and the distribution of $\mathsf{View}(\mathsf{cph}^*)_{\mathsf{cph}}$ and cph' are computationally distinguishable, where $\mathsf{cph}' \leftarrow \mathsf{Update}(\mathsf{cph}, \mathsf{R}^*)$ produced by the challenger, and $\mathsf{View}(\mathsf{cph}^*)_{\mathsf{cph}}$ is a random variable representing the challenger's view regarding the Update algorithm with input cph and R^*.

Definition 4. A drvuKPABE scheme achieves update verifiability if the probability of $\mathcal{A}$ winning the verifiability game is negligible at most with respect to the security parameter ℓ.

4. Main construction

High level idea. In order to construct drvuKPABE, there are mainly two challenges to be resolved: (i) how to make the revoked users' credentials invalid without affecting other non-revoked users' credentials? and (ii) how to allow any third party (e.g., storage provider) to update ciphertexts with public information (i.e., publicly known revocation list)?

In order to resolve the first challenge, the intuition is to separate a user's decryption key into two components, where one is related to access control policy, and the other one is related to the user's identity (meaning that the secret value should be divided into two shares, which are distributed to two respective components). In addition, the ciphertext is composed of two parts, where one is related to the attribute set, and the other is associated to the revocation list. Intuitively, only the attribute set satisfies the access control policy and the user's identity is beyond the revocation list, the plaintext underlying the ciphertext can be recovered. The importance thing here is how to represent that the user's identity is or is not in the revocation list. This naturally leads to the adoption of the subset cover technique. Generally speaking, given the revocation list R, the subset cover $\mathsf{cover}(\mathsf{R})$ can be used to represent non-revoked users (i.e., the leaves that are rooted at nodes of the cover set). Therefore, combining with the multilinear mappings, the second component of the decryption key and second part of the ciphertext can be used to be computed together to recover some secret (indeed it is a function of the secret share generated by the trusted authority).

In order to solve the second challenge, we utilize the one-way property of multilinear mappings, so that original ciphertexts are allowed to be updated as long as the revocation list changes. This works because nodes of the new subset cover (with respect to the new revocation list) are children of nodes in the prior subset cover (with respect to prior revocation). Note here we only consider the case that the user is changed from non-revoked status to revoked status. In this paper we cannot deal with the case that revoked users are changed to the non-revoked status.

drvuKPABE *Construction* we encode an access control policy with (M,π), where M is an $l\times k$ matrix. Let max be the maximum size of attributes allowed to associated with a ciphertext. Let 2^d be the number of users and U be the user universe in the system so that $|\mathsf{U}|=2^d$ and the depth for all leaves in the full binary tree T is d. The drvuKPABE scheme can be constructed as follows:

Setup(1^ℓ): this algorithm initializes the public parameter and master key as follows:

- It generates a $d+2$ multilinear map: $\{e_i: G_0\times G_i\to G_{i+1}|i=0,\ldots,d+1\}$, where $(G_0,G_1,\ldots,G_{d+2})$ are cyclic group of order p respectively. Let g_0 be a generator of G_0, such that $g_{i+1}=e_i(g_0,g_i)$ is the generator of G_{i+1} for $i=0,\ldots,d+1$. Let $\alpha,\beta\overset{R}{\leftarrow}\mathbb{Z}_p$.
- Let H_1,H_2 be two secure hash functions modeled as random oracles, such that $H_1:\{0,1\}^*\to\mathbb{Z}_p$ and $H_2:\{0,1\}^*\to G_0$.
- Let max be the maximum number of attributes that can be associated to a ciphertext, and select $h_j\overset{R}{\leftarrow}G_{d+1},0\leqslant j\leqslant\max$. Given $y\in\mathbb{Z}_p$, define the function $Q(y)=\prod_{j=0}^{\max}\left(h_j^{y^j}\right)$.

It sets the public parameter and master key as

$$\mathsf{pm}=(e_0,\ldots,e_{d+1},G_0,\ldots,G_{d+2},g_0,\ldots,g_{d+2},e_{d+1}(g_0,g_{d+1})^\alpha,g_0^\beta,H_1,H_2,h_0,\ldots,h_{\max}),$$
$$\mathsf{mk}=(\alpha,\beta).$$

$\mathsf{KeyGen}(\mathsf{mk},(\mathsf{M},\pi),\mathsf{uid})$: given the user identity $\mathsf{uid}\in\mathsf{U}$, suppose $\mathsf{path}(\mathsf{uid})=\{x_{i_0},\ldots,x_{i_d}\}$ in the full binary tree T such that $x_{i_0}=root$ and $x_{i_d}=\mathsf{uid}$ (note that the height of the full binary tree is equal to the number of mappings in multilinear maps. Therefore the maximum number of users in the system is 2^d). The decryption key for user uid can be generated as follows:

- Select $\alpha_1,v_2,v_3,\ldots,v_k\overset{R}{\leftarrow}\mathbb{Z}_p$, and set α_2 such that $\alpha=\alpha_1+\alpha_2 \bmod p$,
- Let $\mathbf{v}=(\alpha_1,v_2,\ldots,v_k)$. For $i=1,\ldots,l$ (note that M is an $l\times k$ matrix), compute $\lambda_{\pi(i)}=\mathsf{M}_i\cdot\mathbf{v}$, and set $D_i^{(1)}=g_{d+1}^{\lambda_{\pi(i)}}Q(H_1(\pi(i)))^{r_i},D_i^{(2)}=g_0^{r_i}$ by selecting $r_i\overset{R}{\leftarrow}\mathbb{Z}_p$.
- Given $\mathsf{path}(\mathsf{uid})$, let $P_{x_{i_0}}=e_0(H_2(x_{i_0}),g_0^\beta)$, and then compute $P_{x_{i_j}}=e_j(H_2(x_{i_j}),P_{x_{i_{j-1}}})$ for $j=1,\ldots,d$. Let $D^{(3)}=g_{d+1}^{\alpha_2}P_{\mathsf{uid}}^t,D^{(4)}=g_0^t$ by selecting $t\overset{R}{\leftarrow}\mathbb{Z}_p$.

The decryption key is set to

$$\mathsf{sk}=\left(\mathsf{uid},(\mathsf{M},\pi),\left(D_i^{(1)},D_i^{(2)}\right)_{i\in[1,l]},D^{(3)},D^{(4)}\right).$$

$\mathsf{Enc}(\mathsf{m},\mathsf{S},\mathsf{R})$: the message $\mathsf{m}\in G_{d+2}$ can be encrypted as follows:

- Select $s\overset{R}{\leftarrow}\mathbb{Z}_p$ and set $C^{(1)}=\mathsf{m}e_{d+1}(g_0,g_{d+1})^{\alpha s},C^{(2)}=g_0^s$.
- Given an attribute $\mathsf{at}\in\mathsf{S}$, let $C_{\mathsf{at}}^{(3)}=Q(H_1(\mathsf{at}))^s$.
- Suppose $\mathsf{cover}(\mathsf{R})$ is the cover set with respect to the revocation list R. Given $x\in\mathsf{cover}(\mathsf{R})$, let $\mathsf{path}(x)=\{x_{i_0},\ldots,x_{i_{\mathsf{depth}(x)}}\}$ such that $x_{i_0}=root$ and $x_{i_{\mathsf{depth}(x)}}=x$. Let $P_{x_{i_0}}=e_0(H_2(x_{i_0}),g_0^\beta)$. For $j=1,\ldots,\mathsf{depth}(x)$, compute $P_{x_{i_j}}=e_j(H_2(x_{i_j}),P_{x_{i_{j-1}}})$, and set $C_x^{(4)}=P_x^s$.

The ciphertext is set to

$$\mathsf{cph}=\left(\mathsf{S},\mathsf{R},C^{(1)},C^{(2)},\left\{C_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}},\left\{C_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R})}\right).$$

$\mathsf{Dec}(\mathsf{cph},\mathsf{sk})$: given cph and sk, the decryption can be done as follows:

- If either the user's identity $\mathsf{uid}\in\mathsf{R}$ or the attribute set S does not satisfy the access control policy specified by (M,π), then return $\perp$. Otherwise, proceed as follows.
- Since $\mathsf{uid}\notin\mathsf{R}$, there always exists a node x such that $x\in(\mathsf{path}(\mathsf{uid})\cap\mathsf{cover}(\mathsf{R}))$. Suppose $\mathsf{path}(\mathsf{uid})=\{x_{i_0},\ldots,x_{i_{\mathsf{depth}(x)}},\ldots,x_{i_d}\}$ where $x_{i_{\mathsf{depth}(x)}}=x$ and $x_{i_d}=\mathsf{uid}$. Let $P'_{x_{i_{\mathsf{depth}}(x)}}=C_x^{(4)}$ and compute $P'_{x_{i_{j+1}}}=e_{j+1}\left(H_2(x_{i_{j+1}}),P'_{x_{i_j}}\right)$ for $j=\mathsf{depth}(x),\ldots,d-1$. Eventually, $P'_{\mathsf{uid}}=P'_{x_{i_d}}$.
- Since the attribute set S satisfying the access control policy specified by (M,π), there exists c_i's such that $\sum_{\pi(i)\in\mathsf{S}}c_i\mathsf{M}_i=(1,0,\ldots,0)$, and having

$$K = \prod_{\pi(i)\in \mathsf{S}} \left(\frac{e_{d+1}(C^{(2)}, D_i^{(1)})}{e_{d+1}(D_i^{(2)}, C_{\pi(i)}^{(3)})} \right)^{c_i} \cdot \frac{e_{d+1}(C^{(2)}, D^{(3)})}{e_{d+1}\left(D^{(4)}, P'_{\mathsf{uid}}\right)}.$$

The message can be obtained by computing $\mathsf{m} = C^{(1)}/K$.

$\mathsf{Update}(\mathsf{cph}, \mathsf{R}')$: given a new revocation list R', cph can be updated as follows: suppose $\mathsf{cover}(\mathsf{R}')$ and $\mathsf{cover}(\mathsf{R}')$ are the cover sets with respect to the revocation lists R' and R, respectively. Given $x' \in \mathsf{cover}(\mathsf{R}')$,

- If there exists $x \in \mathsf{cover}(\mathsf{R})$ such that $x = x'$, then set $\widetilde{C}_{x'}^{(4)} = C_x^{(4)}$.
- Otherwise, there exists $x \in \mathsf{cover}(\mathsf{R})$ such that x is an ancestor of x'. Let $\mathsf{path}(x') = \mathsf{path}(x) \cup \{x_{i_{\mathsf{depth}(x)+1}}, \ldots, x_{\mathsf{depth}(x')}\}$ where $x_{i_{\mathsf{depth}(x)}} = x$ and $x_{i_{\mathsf{depth}(x')}} = x'$, and set $P'_{x_{i_{\mathsf{depth}(x)}}} = C_x^{(4)}$. For $j = \mathsf{depth}(x), \ldots, \mathsf{depth}(x') - 1$, compute $P'_{x_{i_{j+1}}} = e_{j+1}(H_2(x_{i_{j+1}}), P'_{x_{i_j}})$. Eventually, set $\widetilde{C}_{x'}^{(4)} = P'_{x'}$.
- Let $\widetilde{C}^{(1)} = C^{(1)}, \widetilde{C}^{(2)} = C^{(2)}, \widetilde{C}_{\mathsf{at}}^{(3)} = C_{\mathsf{at}}^{(3)}$.

The updated ciphertext is set to

$$\mathsf{cph}' = \left(\mathsf{S}, \mathsf{R}', \widetilde{C}^{(1)}, \widetilde{C}^{(2)}, \left\{ \widetilde{C}_{\mathsf{at}}^{(3)} \right\}_{\mathsf{at}\in \mathsf{S}}, \left\{ \widetilde{C}_{x'}^{(4)} \right\}_{x' \in \mathsf{cover}(\mathsf{R}')} \right).$$

$\mathsf{Verify}(\mathsf{cph}, \mathsf{cph}')$: the verification of updating cph' from cph can be done as follows:

- Verify whether the following equations hold simultaneously:

$$C^{(1)} = \widetilde{C}^{(1)}, \quad C^{(2)} = \widetilde{C}^{(2)},$$
$$\forall \mathsf{at} \in \mathsf{S}, \quad C_{\mathsf{at}}^{(3)} = \widetilde{C}_{\mathsf{at}}^{(3)},$$
$$\forall x \in \mathsf{cover}(\mathsf{R}) \cap \mathsf{cover}(\mathsf{R}'), \quad C_x^{(4)} = \widetilde{C}_x^{(4)}.$$

  If not, then output 0. Otherwise, proceed to the following step.
- For $i = 1, \ldots, d$, figure out nodes $x'_1, \ldots, x'_\eta$ such that $x'_j \in \mathsf{cover}(\mathsf{R}') - \mathsf{cover}(\mathsf{R})$ and $\mathsf{depth}(x'_j) = i$, select $c_1, \ldots, c_\eta \xleftarrow{R} \mathbb{Z}_p$, and verify

$$e_{\mathsf{depth}(x')+1}\left(C^{(2)}, \prod_{i=1}^{\eta} P_{x'_i}^{c_i} \right) = e_{\mathsf{depth}(x')+1}\left(g_0, \prod_{i=0}^{\eta} \left(\widetilde{C}_{x'_i}^{(4)} \right)^{c_i} \right).$$

  If there exists $i, 1 \leqslant i \leqslant d$, such that the above equation does not hold, then output 0. Otherwise, return 1.

The correctness of the decryption can be verified as follows:

$$K' = \prod_{\pi(i)\in \mathsf{S}} \left(\frac{e_{d+1}(C^{(2)}, D_i^{(1)})}{e_{d+1}(D_i^{(2)}, C_{\pi(i)}^{(3)})} \right)^{c_i} = \prod_{\pi(i)\in \mathsf{S}} \left(\frac{e_{d+1}\left(g_0^s, g_{d+1}^{\lambda_{\pi(i)}} Q(\pi(i))^{r_i}\right)}{e_{d+1}\left(g_0^{r_i}, Q(\pi(i))^s\right)} \right)^{c_i} = \prod_{\pi(i)\in \mathsf{S}} e_{d+1}\left(g_0^s, g_{d+1}^{\lambda_{\pi(i)}}\right)^{c_i} = e_{d+1}(g_0, g_{d+1})^{\sum_{\pi(i)\in \mathsf{S}} c_i \lambda_{\pi(i)} s}$$
$$= e_{d+1}(g_0, g_{d+1})^{\alpha_1 s},$$

$$K'' = \frac{e_{d+1}(C^{(2)}, D^{(3)})}{e_{d+1}(D^{(4)}, P'_d)} = \frac{e_{d+1}\left(g_0^s, g_{d+1}^{\alpha_2} P_{\mathsf{uid}}^t\right)}{e_{d+1}\left(g_0^t, P_{\mathsf{uid}}^s\right)} = e_{d+1}(g_0, g_{d+1})^{\alpha_2 s},$$

$$K = K'K'' = e_{d+1}(g_0, g_{d+1})^{\alpha_1 s} e_{d+1}(g_0, g_{d+1})^{\alpha_2 s} = e_{d+1}(g_0, g_{d+1})^{\alpha s}.$$

Therefore, we have $C^{(1)}/K = \mathsf{m} e_{d+1}(g_0, g_{d+1})^{\alpha s} / e_{d+1}(g_0, g_{d+1})^{\alpha s} = \mathsf{m}$

5. Security analysis

In the following: we show that the proposed scheme achieves selective security against chosen-plaintext attack on original ciphertext, selective security against chosen-plaintext attacks on updated ciphertext and update verifiability, respectively.

Theorem 1. *Assume that $(d+3)$-MDDH assumption holds, the proposed scheme achieves selective security against chosen-plaintext attack on original ciphertext in the random oracle model.*

Proof. We prove it by showing that if there exists a probabilistic polynomial time adversary $\mathcal{A}$ breaking the selective security game on original ciphertext with advantage ϵ, then we can construct a challenger solving $(d+3)$-MDDH problem with advantage $\epsilon/2$.

Given an instance of $(d+3)$-MDDH problem $(g_0, g_0^{z_0}, \ldots, g_0^{z_d}, g_0^a, g_0^b, g_0^c, Z)$ where $g_0, g_0^{z_0}, \ldots, g_0^{z_d}, g_0^a, g_0^b, g_0^c \xleftarrow{R} G_0, Z \xleftarrow{R} G_{d+2}$ and $z_0, \ldots, z_d, a, b, c$ are unknown, the challenger simulates the game as follows:

Setup: $\mathcal{A}$ selects an attribute set S^* and a revocation list R^*, and sends them to the challenger. The challenger generates the public parameters and master key as follows:

- Given the attribute set S^*, let $\phi(y) = y^{\max - |\mathsf{S}^*|} \cdot \prod_{\mathsf{at} \in \mathsf{S}^*} (y - H_1(\mathsf{at}))$. By being expanded, it can be written as $\phi(y) = \sum_{j=0}^{\max} \phi_j y^j$ where ϕ_j is the coefficient of y^j and therefore $\phi_j = 0$ for $j = 0, \ldots, \max - |\mathsf{S}^*|$.
- Select $\varphi_0, \ldots, \varphi_{\max} \xleftarrow{R} \mathbb{Z}_p$, and define $\varphi(y) = \sum_{j=0}^{\max} \varphi_j y^j$.
- Apply multilinear mapping on $g_0^{z_0}, \ldots, g_0^{z_d}, g_0^a$ to obtain $g_{d+1}^{a'}$ so that a' is implicitly set to $z_0 z_1 \ldots z_d a$.
- Let $h_j = g_{d+1}^{a'\phi_j + \varphi_j}, 0 \leqslant j \leqslant \max$, and define $Q(y) = g_{d+1}^{a'\phi(y) + \varphi(y)} = \prod_{j=0}^{\max} g_{d+1}^{(a'\phi_j + \varphi_j) y^j}$.
- Select $z \xleftarrow{R} \mathbb{Z}_p$, and set the public parameters as

$$\mathsf{pm} = \left(e_0, \ldots, e_{d+1}, g_0, \ldots, g_{d+1}, G_0, \ldots, G_{d+2}, e_{d+1}(g^b, g_{d+1}^{a'}), g_0^{b+z}, H_1, H_2, h_0, \ldots, h_{\max}\right)$$

by implicitly setting the master key $\alpha = a'b$ and $\beta = b + z$.

Moreover, given the revocation list R^*, let $\chi_{\mathsf{R}^*} = \{x \in \mathsf{path}(\mathsf{uid}) | \mathsf{uid} \in \mathsf{R}^*\}$, and H_1, H_2 are simulated as follows:

- $\mathcal{O}_{H_1}(\mathsf{at})$: if at has not been queried before, select $u \xleftarrow{R} \mathbb{Z}_p$, set $H_1(\mathsf{at}) = u$, and add $(\mathsf{at}, H_1(\mathsf{at}))$ to the list L_{H_1}. Otherwise, retrieve $H_1(\mathsf{at})$ from L_{H_1} with respect to at. It returns $H_1(\mathsf{at})$.
- $\mathcal{O}_{H_2}(x)$: given the label x that can be either the label for inner node or user identity with respect to leaf, the random oracle works as follows:
  In the case of $x \in \chi_{\mathsf{R}^*}$: If x has not been queried before, select $v \xleftarrow{R} \mathbb{Z}_p$, set $H_2(x) = g_0^{z_{\mathsf{depth}(x)}} g_0^v$, and add $(x, v, H_2(x))$ to the list L_{H_2}. Otherwise, retrieve $H_2(x)$ from L_{H_2} with respect to x.
  In the case of $x \notin \chi_{\mathsf{R}^*}$: If x has not been queried before, select $v \xleftarrow{R} \mathbb{Z}_p$, set $H_2(x) = g_0^v$, and add $(x, v, H_2(x))$ to the list L_{H_2}. Otherwise, retrieve $H_2(x)$ from L_{H_2} with respect to x.
  It returns $H_2(x)$.

Phase 1: $\mathcal{A}$ can query the oracle in polynomial many times:

- $\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{uid}, \mathsf{P})$: given user's identity uid and the access control policy P specified by (M, π), the challenger proceeds as follows:
  In the case of $\mathsf{uid} \notin \mathsf{R}^*$ and $F(\mathsf{S}^*, P) = 1$ simultaneously, then abort.
  In the case of $\mathsf{uid} \in \mathsf{R}^*$, it selects $\alpha_1 \xleftarrow{R} \mathbb{Z}_p$ and computes $D_i^{(1)}, D_i^{(2)}$ as that specified by algorithm KeyGen. Suppose $\mathsf{path}(\mathsf{uid}) = (x_0, \ldots, x_d = \mathsf{uid})$, and $x_j \in \chi_{\mathsf{R}^*}$ and $H_2(x_j) = g_0^{z_j + v_j}$ since $\mathsf{uid} \in \mathsf{R}^*$. It computes $P_{\mathsf{uid}} = g_{d+1}^{(\sum_{j=0}^{d} (z_j + v_j))(b+z)}$ by applying mappings on $g_0^{z_0 + v_0}, \ldots, g_0^{z_d + v_d}, g_0^{b+z}$. In addition, it chooses $t' \xleftarrow{R} \mathbb{Z}_p$, and sets

$$D^{(3)} = g_{d+1}^{(z_0 z_1 \ldots z_d ab - \alpha_1) + \left(\sum_{j=0}^{d} (z_j + v_j)\right)(b+z)(t'-a)}$$
$$D^{(4)} = g_0^{t'-a} = g_0^{t'} / g_0^a$$

by implicitly defining $\alpha_2 = \alpha - \alpha_1 = z_0 \ldots z_d ab - \alpha_1$ and $t = t' - a$.
In the case of $F(\mathsf{S}^*, \mathsf{P}) = 0$, it chooses $\alpha_2 \xleftarrow{R} \mathbb{Z}_p$ and computes $(D^{(3)}, D^{(4)})$ as that specified by algorithm KeyGen. Since S^* does not satisfy the access structure (M, π), there exists a vector $\mathbf{w} = (w_1, \ldots, w_k) \in \mathbb{Z}_p^k$ such that $w_1 = 1$ and $\forall \pi(i) \in \mathsf{S}^*, \mathsf{M}_i \cdot \mathbf{w} = 0$. It chooses $v_i' \xleftarrow{R} \mathbb{Z}_p$ for $i = 2, \ldots, k$, and sets $\mathbf{v}' = (0, v_2', \ldots, v_k')$. By implicitly setting $\mathbf{v} = (\alpha - \alpha_2)\mathbf{w} + \mathbf{v}'$, it generates $D_i^{(1)}$ and $D_i^{(2)}$ as follows:
  - If $\pi(i) \in \mathsf{S}^*$: it selects $r_i \xleftarrow{R} \mathbb{Z}_p$, and computes $\lambda_{\pi(i)} = \mathsf{M}_i \cdot \mathbf{v}', D_i^{(1)} = g_{d+1}^{\lambda_{\pi(i)}} Q(H_1(\pi(i)))^{r_i}$ and $D_i^{(2)} = g_0^{r_i}$.
  - Otherwise $\pi(i) \notin \mathsf{S}^*$: it selects $r_i' \xleftarrow{R} \mathbb{Z}_p$ and computes

$$D_i^{(1)} = g_{d+1}^{\lambda_{\pi(i)}} Q(H_1(\pi(i)))^{r_i} = g_{d+1}^{\mathsf{M}_i \cdot [(\alpha - \alpha_2)\mathbf{w} + \mathbf{v}']} \cdot g_{d+1}^{[a'\phi(H_1(\pi(i))) + \varphi(H_1(\pi(i)))] r_i} = \frac{g_{d+1}^{\mathsf{M}_i \cdot \mathbf{v}' + a'\phi(H_1(\pi(i)))r_i' + \varphi(H_1(\pi(i)))r_i'}}{g_{d+1}^{\alpha_2 \mathsf{M}_i \cdot \mathbf{w} + b\varphi(H_1(\pi(i)))\mathsf{M}_i \cdot \mathbf{w}/\phi(H_1(\pi(i)))}}$$

$$D_i^{(2)} = g_0^{-b\mathsf{M}_i \cdot \mathbf{w}/\phi(H_1(\pi(i))) + r_i'}$$

by implicitly setting $r_i = r_i' - b\mathsf{M}_i \cdot \mathbf{w}/\phi(H_1(\pi(i)))$.

Challenge: $\mathcal{A}$ selects two messages $\mathsf{m}_0, \mathsf{m}_1$ of equal-length and sends them to the challenger. The challenger chooses $\sigma \xleftarrow{R} \{0,1\}$, and computes the ciphertext as follows:

- Given every attribute $\mathsf{at} \in \mathsf{S}^*$, it computes $C^{(1)} = \mathsf{m}_\sigma Z, C^{(2)} = g_0^c$ and $C_{\mathsf{at}}^{(3)} = g_{d+1}^{c\varphi(H_1(\mathsf{at}))}$.
- Given every $x \in \mathsf{cover}(\mathsf{R}^*)$, suppose $\mathsf{path}(x) = (x_0, \ldots, x_{\mathsf{depth}(x)} (= x))$, then $x_i \in \chi_{\mathsf{R}^*}, i = 0, \ldots, \mathsf{depth}(x) - 1$, and $x \notin \chi_{\mathsf{R}^*}$. It sets

$$C_x^{(4)} = P_{\mathsf{depth}(x)}^c = g_{\mathsf{depth}(x)+1}^{(\sum_{j=0}^{\mathsf{depth}(x)-1}(z_i+v_i))v_{\mathsf{depth}(x)}(b+z)c}.$$

The challenger sends to $\mathcal{A}$

$$\mathsf{cph}^* = \left(C^{(1)}, C^{(2)}, \left\{C_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}^*}, \left\{C_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R})}\right).$$

Guess: $\mathcal{A}$ outputs a guess σ'. The challenger outputs $Z = g_{d+2}^{z_0\cdots z_d abc}$ if $\sigma' = \sigma$. Otherwise, it outputs $Z \neq g_{d+2}^{z_0\cdots z_d abc}$.

This completes the simulation. In the challenge phase, if $Z = g_{d+2}^{z_0\cdots z_d abc}$, then cph^* is indeed a valid ciphertext for m_σ. Then the probability of $\mathcal{A}$ outputting $\sigma = \sigma'$ is $\frac{1}{2} + \epsilon$. If Z is an element randomly selected from G_{d+2}, the probability of $\mathcal{A}$ outputting $\sigma = \sigma'$ is $\frac{1}{2}$. Therefore, the probability of the challenger correctly guessing $Z \stackrel{?}{=} g_{d+2}^{z_0\cdots z_d abc}$ is $\frac{1}{2}\left(\frac{1}{2}+\epsilon\right) + \frac{1}{2}\frac{1}{2} = \frac{1}{2} + \frac{\epsilon}{2}$. That is, the challenger solves the $(d+3)$-MDDH problem with advantage $\epsilon/2$ if $\mathcal{A}$ wins the selective security game on original ciphertext with advantage ϵ. □

Theorem 2. *Given the $(d+3)$-MDDH assumption, the proposed scheme achieves selective security on updated ciphertext in the random oracle model.*

Proof. In Theorem 1, we show that any probabilistic polynomial time adversary $\mathcal{A}$ cannot learn any information from original ciphertexts if either he is in the revocation list (i.e., $\mathsf{uid} \in \mathsf{R}$) or the attribute set specified by the ciphertext does not match the access control policy specified by his decryption key. The strategy of proving this theorem is to argue that if the distribution of ciphertexts generated by algorithm Enc is identical to that of the ciphertexts generated by algorithm Update, then the proposed scheme achieves selective security on updated ciphertext. This argument is valid because we can construct a simulator using the adversary $\mathcal{A}$ to break selective security on original ciphertext if it does not hold.

Now let us look at the distributions of original ciphertexts and updated ciphertexts, respectively, given the same message m, attributes set S, and the revocation list R^* (Suppose R^* is changed from R, such that $\mathsf{R} \subset \mathsf{R}^*$):

- The original ciphertext generated by algorithm $\mathsf{Enc}(\mathsf{m}, \mathsf{S}, \mathsf{R}^*)$ is:

$$\mathsf{cph}^* = \left(\mathsf{S}, \mathsf{R}^*, C^{*(1)}, C^{*(2)}, \left\{C_{\mathsf{at}}^{*(3)}\right\}_{\mathsf{at}\in\mathsf{S}}, \left\{C_{x'}^{*(4)}\right\}_{x'\in\mathsf{cover}(\mathsf{R}^*)}\right),$$

where $C^{*(1)} = \mathsf{m}e_{d+1}(g_0, g_{d+1})^{\alpha s}, C^{*(2)} = g_0^s, C_{\mathsf{at}}^{*(3)} = Q(H_1(\mathsf{at}))^s, C_{x'}^{*(4)} = P_{x'}^s$.

- The original ciphertext generated by algorithm $\mathsf{Enc}(\mathsf{m}, \mathsf{S}, \mathsf{R})$ is:

$$\mathsf{cph} = \left(\mathsf{S}, \mathsf{R}, C^{(1)}, C^{(2)}, \left\{C_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}}, \left\{C_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R})}\right),$$

where $C^{(1)} = \mathsf{m}e_{d+1}(g_0, g_{d+1})^{\alpha s'}, C^{(2)} = g_0^{s'}, C_{\mathsf{at}}^{(3)} = Q(H_1(\mathsf{at}))^{s'}, C_x^{(4)} = P_x^{s'}$.

Then the updated ciphertext generated by algorithm $\mathsf{Update}(\mathsf{cph}, \mathsf{R}^*)$ is:

$$\mathsf{cph}' = \left(\mathsf{S}, \mathsf{R}^*, \widetilde{C}^{(1)}, \widetilde{C}^{(2)}, \left\{\widetilde{C}_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}}, \left\{\widetilde{C}_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R}^*)}\right),$$

where $\widetilde{C}^{(1)} = \mathsf{m}e_{d+1}(g_0, g_{d+1})^{\alpha s'}, \widetilde{C}^{(2)} = g_0^{s'}, \widetilde{C}_{\mathsf{at}}^{(3)} = Q(H_1(\mathsf{at}))^{s'}$. For all $x' \in \mathsf{cover}(\mathsf{R}) \cap \mathsf{cover}(\mathsf{R}^*), \widetilde{C}_{x'}^{(4)} = P_{x'}^{s'}$, and for all $x' \in \mathsf{cover}(\mathsf{R}^*) - \mathsf{cover}(\mathsf{R}), \widetilde{C}_{x'}^{(4)} = P'_{x'} = P_{x'}^{s'}$.

We can see that the original ciphertext (i.e., cph^*) and updated ciphertext (i.e., cph') have exactly the same number of terms. In addition, corresponding terms are randomized with random values s and s', respectively. Therefore, original ciphertexts and updated ciphertexts have the identical distribution, meaning that $\mathcal{A}$ cannot distinguish whether the ciphertext is generated from Enc algorithm or Update algorithm. That is, the advantage of $\mathcal{A}$ in the selective security games on updated ciphertext is as same as that of winning the selective security game on original ciphertext. Hence, if the adversary breaks the selective security on updated ciphertext, then it can break the selective security game on original ciphertext, which therefore solves the $(d+3)$-MDDH problem. □

Theorem 3. *The proposed scheme achieves verifiability.*

Proof. Suppose cph is the original ciphertext of message m with respect to revocation list R, denoted by

$$\mathsf{cph} = \left(\mathsf{S}, \mathsf{R}, C^{(1)}, C^{(2)}, \left\{C_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}}, \left\{C_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R})}\right).$$

Let cph^* be the updated ciphertext returned by the adversary $\mathcal{A}$, when the revocation list R is changed to R^*, such that $\mathsf{R}\subset\mathsf{R}^*$, denoted by

$$\mathsf{cph}^* = \left(\mathsf{S}, \mathsf{R}^*, \widetilde{C^*}^{(1)}, \widetilde{C^*}^{(2)}, \left\{\widetilde{C^*}_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}}, \left\{\widetilde{C^*}_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R}^*)}\right).$$

Suppose cph' is the updated ciphertext output by algorithm Update, denoted by

$$\mathsf{cph}' = \left(\mathsf{S}, \mathsf{R}^*, \widetilde{C}'^{(1)}, \widetilde{C}'^{(2)}, \{\widetilde{C}'^{(3)}_{\mathsf{at}}\}_{\mathsf{at}\in\mathsf{S}}, \left\{\widetilde{C}'^{(4)}_x\right\}_{x\in\mathsf{cover}(\mathsf{R}^*)}\right).$$

We claim that $\mathsf{cph}' = \mathsf{cph}^*$ in order to attain $1 \leftarrow \mathsf{Verify}(\mathsf{cph}, \mathsf{cph}^*)$ (note $1 \leftarrow \mathsf{Verify}(\mathsf{cph}, \mathsf{cph}')$ due to the correctness of the scheme). Suppose that $\mathsf{Verify}(\mathsf{cph}, \mathsf{cph}^*)$ outputs 1, then the following should hold:

$$\begin{aligned}
&C^{(1)} = \widetilde{C^*}^{(1)}, \quad C^{(2)} = \widetilde{C^*}^{(2)},\\
&\left\{C_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}} = \left\{\widetilde{C^*}_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}},\\
&\left\{C_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R})\cap\mathsf{cover}(\mathsf{R}^*)} = \left\{\widetilde{C^*}_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R})\cap\mathsf{cover}(\mathsf{R}^*)}
\end{aligned}$$

and

$$\forall j = 1,\ldots,d, e_{j+1}\left(C^{(2)}, \prod_{i=1}^{\eta} P_{x_i'}^{c_i}\right) = e_{j+1}\left(g_0, \prod_{i=1}^{\eta}\left(\widetilde{C^*}_{x_i'}^{(4)}\right)^{c_i}\right), \tag{1}$$

where $\left\{x_1', x_2', \ldots, x_\eta'\right\} = \mathsf{cover}(\mathsf{R}^*) - \mathsf{cover}(\mathsf{R})$ are the nodes of depth j and $c_1,\ldots,c_\eta \in \mathbb{Z}_p$ are randomly selected by the challenger and unknown to $\mathcal{A}$.

Since $1 \leftarrow \mathsf{Verify}(\mathsf{cph}, \mathsf{cph}')$, the following should hold:

$$\begin{aligned}
&C^{(1)} = \widetilde{C}'^{(1)}, C^{(2)} = \widetilde{C}'^{(2)},\\
&\left\{C_{\mathsf{at}}^{(3)}\right\}_{\mathsf{at}\in\mathsf{S}} = \left\{\widetilde{C}'^{(3)}_{\mathsf{at}}\right\}_{\mathsf{at}\in\mathsf{S}},\\
&\left\{C_x^{(4)}\right\}_{x\in\mathsf{cover}(\mathsf{R})\cap\mathsf{cover}(\mathsf{R}^*)} = \left\{\widetilde{C}'^{(4)}_x\right\}_{x\in\mathsf{cover}(\mathsf{R})\cap\mathsf{cover}(\mathsf{R}^*)}
\end{aligned}$$

and

$$\forall j = 1,\ldots,d, e_{j+1}\left(C^{(2)}, \prod_{i=1}^{\eta} P_{x_i'}^{c_i}\right) = e_{j+1}\left(g_0, \prod_{i=1}^{\eta}\left(\widetilde{C}'^{(4)}_{x_i'}\right)^{c_i}\right) \tag{2}$$

where $\left\{x_1', x_2', \ldots, x_\eta'\right\} = \mathsf{cover}(\mathsf{R}^*) - \mathsf{cover}(\mathsf{R})$ are the nodes of depth j and $c_1,\ldots,c_\eta \in \mathbb{Z}_p$ are random values the same as above (Note that $\forall j, \left\{x_1', x_2', \ldots, x_\eta'\right\} = \mathsf{cover}(\mathsf{R}^*) - \mathsf{cover}(\mathsf{R})$ of depth j are the same as above because R, R^* are public known).

In order to prove $\mathsf{cph}' = \mathsf{cph}^*$, it needs to prove $\forall i, 1 \leqslant i \leqslant \eta, \widetilde{C^*}_{x_i'}^{(4)} = \widetilde{C}'^{(4)}_{x_i'}$. We prove this by showing that the probability of $\widetilde{C^*}_{x_i'}^{(4)} = \widetilde{C}'^{(4)}_{x_i'}$ for some i is negligible.

According to Eq. (1) and (2), the following holds

$$e_{j+1}\left(g_0, \prod_{i=1}^{\eta}\left(\widetilde{C}'^{(4)}_{x_i'}\right)^{c_i}\right) = e_{j+1}\left(g_0, \prod_{i=1}^{\eta}\left(\widetilde{C^*}_{x_i'}^{(4)}\right)^{c_i}\right).$$

Then we have

$$\prod_{i=1}^{\eta}\left(\widetilde{C}'^{(4)}_{x_i'}\right)^{c_i} = \prod_{i=1}^{\eta}\left(\widetilde{C^*}_{x_i'}^{(4)}\right)^{c_i}.$$

Assume there exists a subset $I\subset[1,\eta]$ such that $\forall t, t\in I, \widetilde{C}_{x_t'}^{(4)} \neq \widetilde{C}'^{(4)}_{x_t'}$. Therefore, we cancel the identical items at both sides and get

$$\prod_{t\in I}\left(\widetilde{C}'^{(4)}_{x_t'}\right)^{c_t} = \prod_{t\in I}\left(\widetilde{C}_{x_t'}^{(4)}\right)^{c_t}.$$

Suppose $\widetilde{C}'^{(4)}_{x'_t} = g_j^{\mu'_t}$ and $\widetilde{C}^{(4)}_{x'_t} = g_j^{\mu_t}$ for some unknown $\mu'_t, \mu_t \in \mathbb{Z}_p$ and $\mu'_t \neq \mu_t$, then we have

$$\prod_{t\in I} g_j^{c_t(\mu'_t-\mu_t)} = 1,$$

meaning that $\sum_{t\in I} c_t(\mu'_t - \mu_t) = 0 \bmod p$. Because c_t is unknown to $\mathcal{A}$, we can see that the probability of $\widetilde{C}^{*}_{x'_t} \neq \widetilde{C}'_{x'_t}$ is at most $1/p$, which is a negligible probability. That is, we show that $\forall i, 1 \leqslant i \leqslant \eta, \widetilde{C}^{(4)}_{x'_i} = \widetilde{C}'^{(4)}_{x'_i}$.

Therefore, we show that $\mathsf{cph}^* = \mathsf{cph}'$ in order to achieve $1 \leftarrow \mathsf{Verify}(\mathsf{cph}, \mathsf{cph}^*)$. That is, the adversary $\mathcal{A}$ cannot present an incorrect updated ciphertext while passing the verification. □

6. Conclusions

In this paper, we propose a novel ABE variant, called directly revocable key-policy ABE with verifiable ciphertext delegation. Our solution can be used in data sharing applications in the cloud setting, allowing (1) the trusted authority to revoke users directly by updating the revocation list without any interaction with non-revoked users, (2) the third party (e.g., the storage provider) to update ciphertexts in order to assure that revoked users cannot decrypt ciphertexts successfully, and (3) any auditor (e.g., authorized by data owners) to verify whether the untrusted third party updated ciphertexts correctly. One of our future work is to construct directly revocable ciphertext-policy ABE with verifiable ciphertext delegation.

Acknowledgment

We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by Program for New Century Excellent Talents in University (NCET-11-0565), the Fundamental Research Funds for the Central Universities (2012JBZ010) and PCSIRT (No. IRT 201206).

References

[1] N. Attrapadung, H. Imai, Attribute-based encryption supporting direct/indirect revocation modes, in: IMA Int. Conf., 2009, pp. 278–300.
[2] N. Attrapadung, H. Imai, Conjunctive broadcast and attribute-based encryption, in: Pairing-Based Cryptography–Pairing 2009, Springer, 2009, pp. 248–265.
[3] A. Balu, K. Kuppusamy, An expressive and provably secure ciphertext-policy attribute-based encryption, Inf. Sci. (2013).
[4] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: IEEE Symposium on Security and Privacy, 2007, pp. 321–334.
[5] A. Boldyreva, V. Goyal, V. Kumar, Identity-based encryption with efficient revocation, in: ACM Conference on Computer and Communications Security, 2008, pp. 417–426.
[6] D. Boneh, A. Silverberg, Applications of multilinear forms to cryptography, Contemp. Math. 324 (2002) 71–90.
[7] M. Chase, Multi-authority attribute based encryption, in: TCC, 2007, pp. 515–534.
[8] J.-S. Coron, T. Lepoint, M. Tibouchi, Practical multilinear maps over the integers, in: CRYPTO, 2013, pp. 476–493.
[9] H. Deng, Q. Wu, B. Qin, J. Domingo-Ferrer, L. Zhang, J. Liu, W. Shi, Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts, Inf. Sci. (2014).
[10] S. Garg, C. Gentry, S. Halevi, Candidate multilinear maps from ideal lattices, in: EUROCRYPT, 2013, pp. 1–17.
[11] V. Goyal, A. Jain, O. Pandey, A. Sahai, Bounded ciphertext policy attribute based encryption, in: ICALP, Part II, Springer-Verlag, 2008, pp. 579–591.
[12] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: ACM Conference on Computer and Communications Security, 2006, pp. 89–98.
[13] M. Green, S. Hohenberger, B. Waters, Outsourcing the decryption of abe ciphertexts, in: USENIX Security Symposium, 2011, p. 3.
[14] J. Lai, R.H. Deng, C. Guan, J. Weng, Attribute-based encryption with verifiable outsourced decryption, IEEE Trans. Inf. Forensic. Secur. 8 (8) (2013) 1343–1354.
[15] H. Lin, Z. Cao, X. Liang, J. Shao, Secure threshold multi authority attribute based encryption without a central authority, Inf. Sci. 180 (13) (2010) 2618–2632.
[16] Q. Liu, G. Wang, J. Wu, Time-based proxy re-encryption scheme for secure data sharing in a cloud environment, Inf. Sci. 258 (2014) 355–370.
[17] Y. Ming, L. Fan, H. Jing-Li, W. Zhao-Li, An efficient attribute based encryption scheme with revocation for outsourced data sharing control, in: 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, IEEE, 2011, pp. 516–520.
[18] D. Naor, M. Naor, J. Lotspiech, Revocation and tracing schemes for stateless receivers, in: CRYPTO, 2001, pp. 41–62.
[19] T. Naruse, M. Mohri, Y. Shiraishi, Attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating, in: Future Information Technology, Springer, 2014, pp. 119–125.
[20] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, ACM, 2007, pp. 195–203.
[21] M. Pirretti, P. Traynor, P. McDaniel, B. Waters, Secure attribute-based systems, in: ACM Conference on Computer and Communications Security, 2006, pp. 99–112.
[22] A. Sahai, H. Seyalioglu, B. Waters, Dynamic credentials and ciphertext delegation for attribute-based encryption, in: CRYPTO, 2012, pp. 199–217.
[23] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: EUROCRYPT, 2005, pp. 457–473.
[24] B. Waters, Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization, in: Public Key Cryptography, 2011, pp. 53–70.
[25] C.K. Wong, M.G. Gouda, S.S. Lam, Secure group communications using key graphs, IEEE ACM Trans. Netw. 8 (1) (2000) 16–30.
[26] Y. Zhang, X. Chen, J. Li, H. Li, F. Li, Fdr-abe: attribute-based encryption with flexible and direct revocation, in: 2013 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS), IEEE, 2013, pp. 38–45.
[27] Q. Zheng, S. Xu, G. Ateniese, Vabks: verifiable attribute-based keyword search over outsourced encrypted data, Cryptology ePrint Archive, Report 2013/462, <http://eprint.iacr.org/>, 2013.