Attribute-Based Encryption Supporting Direct/Indirect Revocation Modes

Nuttapong Attrapadung and Hideki Imai

Research Center for Information Security (RCIS),
National Institute of Advanced Industrial Science and Technology (AIST)
Akihabara-Daibiru Room 1003, 1-18-13, Sotokanda,
Chiyoda-ku, Tokyo 101-0021 Japan
{n.attrapadung,h-imai}@aist.go.jp

Abstract. Attribute-based encryption (ABE) enables an access control mechanism over encrypted data by specifying access policies among private keys and ciphertexts. In this paper, we focus on ABE that supports revocation. Currently, there are two available revocable ABE schemes in the literature. Their revocation mechanisms, however, differ in the sense that they can be considered as direct and indirect methods. *Direct revocation* enforces revocation directly by the sender who specifies the revocation list while encrypting. *Indirect revocation* enforces revocation by the key authority who releases a key update material periodically in such a way that only non-revoked users can update their keys (hence, revoked users' keys are implicitly rendered useless). An advantage of the indirect method over the direct one is that it does not require senders to know the revocation list. In contrast, an advantage of the direct method over the other is that it does not involve key update phase for all non-revoked users interacting with the key authority. In this paper, we present the first *Hybrid Revocable ABE* scheme that allows senders to select on-the-fly when encrypting whether to use either direct or indirect revocation mode; therefore, it combines best advantages from both methods.

1 Introduction

Attribute-based encryption (ABE) enables an access control mechanism over encrypted data using access policies and ascribed attributes among private keys and ciphertexts. ABE was introduced first by Sahai and Waters [20] and refined by many subsequent works [16,5,19,9,15,22,2]. In an ABE system, an encryptor specifies a set of attributes, which could be any keywords describing the ciphertext, directly in the encryption algorithm (which can be run by anyone knowing the universal public key issued priorly by an authority). A user in the system possesses a key associated with an access policy, stating what kind of ciphertext that she can decrypt. Users' keys are priorly given from the key authority. Such a user can decrypt a ciphertext if the policy associated to her key is satisfied by the attribute set associated with the ciphertext. An example application of ABE is pay-TV system with package policy (called target broadcast system

M.G. Parker (Ed.): Cryptography and Coding 2009, LNCS 5921, pp. 278–300, 2009.

in [16]). There, a ciphertext will be associated with an attribute set, such as $\omega = \{$"TITLE:24", "GENRE:SUSPENSE", "SEASON:2", "EPISODE:13" $\}$, while a policy such as $\mathbb{A}$ = "SOCCER" $\vee$ ("TITLE:24" $\wedge$ "SEASON:5") will be associated to TV program package keys that user receives when subscribes.

1.1 Motivation

Revocation mechanism is necessary for any encryption schemes that involve many users, since some private keys might get compromised at some point. In simpler primitives such as public key infrastructure and ID-based encryption (IBE), there are many revocation methods proposed in the literature [17,1,18,7,12,6,4].

In attribute-based setting, Boldyreva et al. [6] only recently proposed a revocable ABE scheme (extended from their main contribution, a revocable IBE). Their scheme uses a key update approach roughly as follows. The sender will encrypt with the attribute set ω as usual, and in addition, he also specifies the present time slot attribute, *e.g.,* "TIME:2009.WEEK49". The key authority, who possesses the current revocation list, periodically announces a key update material at each time slot so that only non-revoked users can update their key and use it to decrypt ciphertexts encrypted at the present time. We call this approach an *indirect* revocation, since the authority indirectly enables revocation by forcing revoked users to be unable to update their keys.

The indirect revocation has an advantage that senders do not need to know the revocation list. However, it also has a disadvantage that the key update phase can be a bottleneck since it requires communication from the key authority to *all* non-revoked users at *all* time slots. One of the main motivations for Boldyreva et al. [6] was also to reduce this cost from a naive approach which would requires the update key of size $O(n - r)$ group elements. Here n is the number of users, r is the number of revoked users. Their scheme reduces this to $O(r \log(\frac{n}{r}))$, by using the classic Complete-subtree method [1,18] combined in a non-trivial way with the fuzzy IBE scheme of [20].

In order to eliminate this bottlenecked key update phase completely, Attrapadung and Imai [3] recently proposed an ABE system with *direct revocation*. Such a system allows senders to specify the revocation list directly when encrypting. Therefore, revocation can be done instantly and does not require the key update phase as in the indirect method. Despite this clear advantage, in contrast, its disadvantage is that it requires senders to possesses the current revocation list. While the management of revocation list itself could be already a troublesome task, this requirement renders the system not being so purely attribute-based. (An ideal attribute-based setting should allow senders to just create ciphertext based solely on attributes and not to worry about revocation). The authors in [3] argued that, however, this setting is still reasonable for some applications such as the Pay-TV example above, where the sender is the TV program distributor company, who should possess the pirate key list to be revoked.

This nature of such an exact opposite advantage tradeoff between the direct and indirect revocation motivates us to look for a more flexible system that supports both revocation methods so that we could have the best of both worlds.

1.2 Our Goal and Contributions

In this paper, we propose a new system called *Hybrid Revocable Attribute-Based Encryption (HR-ABE)*. This system allows a sender Alice to be able to select whether to use either direct or indirect revocation mode on-the-fly when encrypting a message. On the other hand, a user Bob possesses only one key but will be able to decrypt ciphertexts that were constructed in *either modes*.

An HR-ABE works as follows. When Alice selects the direct mode, she will specify the revocation list R directly into the encryption algorithm. On the other hand, when selecting the indirect mode, she is required only to specify the present time slot t (besides the usual attribute set input). A user Bob has one private key. Let $\mathbb{A}$ be the access policy associated to Bob's key. In addition, his key will be associated with a unique serial number id. If ciphertext was from the direct mode, he can decrypt solely by his key. Let ω be the attribute set associated with ciphertext. In this case, he can decrypt if ω satisfies $\mathbb{A}$ and $\mathsf{id} \notin R$. If ciphertext was from indirect mode, he must obtain an update key $\mathsf{uk}_{(R,t)}$ from the authority at time t. Again, he can decrypt if ω satisfies $\mathbb{A}$, and $\mathsf{id} \notin R$. Notice that in this latter case, the key authority specifies R when creating the update key, hence enforces revocation indirectly.

One trivial construction for HR-ABE is to use two sub-systems: a directly revocable ABE and an indirectly revocable ABE. A user key then consists of two keys, one from each sub-system. To encrypt in a desired mode, Alice just uses the corresponding sub-system. The problem for this approach is that the key size will be the sum of key sizes from both sub-systems. Therefore, our goal is to construct an efficient scheme which has the key size being roughly the same as in either the currently best directly or indirectly revocable ABE.

Another goal in designing a scheme is to base its security to the weakest assumption as possible. Since the currently most efficient (non-revocable) ABE [16] is based on the Decision Bilinear Diffie-Hellman (DBDH) assumption, we will also base the security of our scheme on this assumption.

The currently best indirectly revocable ABE that is based on DBDH assumption is the scheme of Boldyreva et al. [6] (given only implicitly in their paper, though). For the case of directly revocable ABE, to the best of our knowledge, no such scheme is available yet. However, in this paper, we give a notice that a variant of Attrapadung-Imai [3] achieves such a property.[1] Both the scheme of [6] and the variant of [3] have the key size of $2\ell \log(n)$ group elements. Therefore, the trivial combination yields the key size of $4\ell \log(n)$. Here ℓ is the number of attributes appear in the policy.

In this paper, we first formalize the notions of HR-ABE and propose a HR-ABE scheme with key size $2(\ell + 1)\log(n)$. This is roughly the same size as the two above one-mode revocable ABEs (and hence half size of the trivial combined scheme). The ciphertext size in direct mode is the same as the variant of [3]. The

[1] This variant itself is not trivial but we jump a step forward to construct a hybrid revocable ABE system, instead of only a directly revocable one. We will mention this variant in Section §6, though.

Table 1. Simple comparison among encryption primitives supporting revocation

	Indirectly revocable ABE	Directly revocable ABE	Broadcast encryption
Attribute-based setting	✓	✓	×
Revocator	Authority	Sender	Sender
No need for key update	×	✓	✓

ciphertext and update key sizes in indirect mode is the same as that of [6]. See Table 2 in Section §6 for comparison.

The security of our scheme is based on the DBDH assumption in the standard model. The security proof is itself quite non-trivial since in the proof, we have to simulate the key of same structure to be able to handle both attack models corresponding to two revocation modes.

1.3 Related Works

Broadcast Encryption. Broadcast encryption (BE) schemes [11,18,8,21,13] allow a sender to specify a receiver group when encrypting. The reader might wonder whether we can just use a public-key BE system instead of ABE in the case when the sender knows the revocation list by simply specifying all non-revoked users as the receiver group. The answer is that we cannot, since we focus on the *attribute-based setting*, which means that the sender is supposed not to even know whose access policy will match the attribute set associated to ciphertext. We provide a simple comparison in Table 1.

Other ABE Variants. The ABE system we have discussed so far is called Key-Policy ABE (KP-ABE) [16]. There is another opposite variant called Ciphertext-Policy ABE (CP-ABE) [5]. In CP-ABE, the roles of an attribute set and an access policy are swapped from what we described for KP-ABE. Attribute sets are associated to keys and access policies over these attributes are associated to ciphertexts. An example application of CP-ABE is secure mailing list system with access policy. There, a private key will be assigned for an attribute set, such as {"MANAGER", "AGE:30", "INSTITUTE:ABC"}, while policies over attributes such as "MANAGER" $\vee$ ("TRAINEE" $\wedge$ "AGE:25") will be associated to ciphertexts.

In this paper, we will consider only the KP-ABE variant. However, the methodology can be applied to the case of CP-ABE similarly.

Previous Works on ABE. ABE was introduced by Sahai and Waters [20] in the context of a generalization of IBE called Fuzzy IBE, which is an ABE that allows only single threshold access structures. The first (and still being state-of-the-art) KP-ABE that allow any monotone access structures was proposed by Goyal et al. [16], while the first such CP-ABE, albeit with the security proof in the generic bilinear group model, was proposed by Bethencourt, Sahai, and Waters [5]. Ostrovsky, Sahai, and Waters [19] then subsequently extended both schemes to handle also any non-monotone structures; therefore, negated clauses

can be specified in policies. Goyal et al. [15] presented bounded CP-ABE in the standard model. Waters [22] recently proposed the first fully expressive CP-ABE in the standard model. Chase [9] presented multi-authority KP-ABE. Recently, Attrapadung and Imai [2] proposed a new ABE variant called Dual-Policy ABE which is a combination of both KP and CP ABE. Revocable ABE was first mentioned by Gollé et al. [14], but their scheme was only heuristic.

2 Preliminaries

2.1 Access Structures and Linear Secret Sharing

We first provide the notion of access structure and linear secret sharing scheme as follows. Such formalization is recapped from [22].

Definition 1 (Access Structures). *Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\mathcal{P}}$ is monotone if for all B, C we have that if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (resp., monotonic access structure) is a collection (resp., monotone collection) $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$.*

Definition 2 (Linear Secret Sharing Schemes (LSSS)). *Let $\mathcal{P}$ be a set of parties. Let M be a matrix of size $\ell \times k$. Let $\pi : \{1, \ldots, \ell\} \to \mathcal{P}$ be a function that maps a row to a party for labeling. A secret sharing scheme Π for access structure $\mathbb{A}$ over a set of parties $\mathcal{P}$ is a linear secret-sharing scheme in $\mathbb{Z}_p$ and is represented by (M, π) if it consists of two polynomial-time algorithms:*

$\mathsf{Share}_{(M,\pi)}$**:** *The algorithm takes as input $s \in \mathbb{Z}_p$ which is to be shared. It randomly chooses $y_2, \ldots, y_k \in \mathbb{Z}_p$ and let $\boldsymbol{v} = (s, y_2, \ldots, y_k)$. It outputs $M\boldsymbol{v}$ as the vector of ℓ shares. The share $\lambda_{\pi(i)} := \boldsymbol{M_i} \cdot \boldsymbol{v}$ belongs to party $\pi(i)$, where we denote $\boldsymbol{M_i}$ as the ith row in M.*

$\mathsf{Recon}_{(M,\pi)}$**:** *The algorithm takes as input $S \in \mathbb{A}$. Let $I = \{i |\ \pi(i) \in S\}$. It outputs reconstruction constants $\{(i, \mu_i)\}_{i \in I}$ which has a linear reconstruction property: $\sum_{i \in I} \mu_i \cdot \lambda_{\pi(i)} = s$.*

Lemma 1. *([22]) Let (M, π) be a LSSS for access structure $\mathbb{A}$ over a set of parties $\mathcal{P}$, where M is a matrix of size $\ell \times k$. For all $S \notin \mathbb{A}$, there exists a polynomial time algorithm that outputs a vector $\boldsymbol{w} = (w_1, \ldots, w_k) \in \mathbb{Z}_p^k$ such that $w_1 = 1$ (or w_1 can be chosen arbitrarily in $\mathbb{Z}_p$) and for all $i \in [1, \ell]$ where $\pi(i) \in S$ it holds that $\boldsymbol{M_i} \cdot \boldsymbol{w} = 0$.*

2.2 Bilinear Maps and Some Assumptions

Bilinear Maps. We briefly review some facts about bilinear maps. Let $\mathbb{G}, \mathbb{G}_T$ be multiplicative groups of prime order p. Let g be a generator of $\mathbb{G}$. A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ for which the following hold: (1) e is bilinear; that is, for all $u, v \in \mathbb{G}$, $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$. (2) The map is non-degenerate: $e(g, g) \neq 1$. We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists $\mathbb{G}_T$ for which the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is efficiently computable.

DBDH Assumption. Let $\mathbb{G}$ be a bilinear group of prime order p. The DBDH (Decision Bilinear Diffie-Hellman) problem [7] in $\mathbb{G}$ is stated as follows. Given a tuple $(g, g^a, g^b, g^s) \in \mathbb{G}^4$ and an element $Z \in \mathbb{G}_T$ as input, determine if $Z = e(g,g)^{abs}$. An algorithm $\mathcal{A}$ that outputs $b \in \{0,1\}$ has advantage ϵ in solving the DBDH problem in $\mathbb{G}$ if $|\Pr[\mathcal{A}(g, g^a, g^b, g^s, e(g,g)^{abs}) = 0] - \Pr[\mathcal{A}(g, g^a, g^b, g^s, Z) = 0]| \geq \epsilon$. We refer to the distribution on the left as $\mathcal{P}_{BDH}$ and the one on the right as $\mathcal{R}_{BDH}$. We say that the DBDH assumption holds in $\mathbb{G}$ if no polynomial-time algorithm has a non-negligible advantage in solving the problem.

2.3 Some Terminologies for Binary Tree

We denote some terminology for complete binary tree. Let $\mathcal{L} = \{1, \ldots, n\}$ be the set of leaves. Let $\mathcal{X}$ be the set of node names in the tree via some systematic naming order. For a leaf $i \in \mathcal{L}$, let $\mathsf{Path}(i) \subset \mathcal{X}$ be the set of all nodes on the path from node i to the root (including i and the root).

For $R \subseteq \mathcal{L}$, let $\mathsf{Cover}(R) \subset \mathcal{X}$ be defined as follows. First mark all the nodes in $\mathsf{Path}(i)$ if $i \in R$. Then $\mathsf{Cover}(R)$ is the set of all the unmarked children of marked nodes. It can be shown to be the minimal set that contains no node in $\mathsf{Path}(i)$ if $i \in R$ but contains at least one node in $\mathsf{Path}(i)$ if $i \notin R$. This function was widely used, e.g., in revocation scheme of [1] and the Complete-Subtree broadcast encryption [18]. It is known [1,18] that $|\mathsf{Cover}(R)| \leq |R|(\log(n/|R|) + 1)$.

2.4 Lagrange Interpolation

For $i \in \mathbb{Z}$ and $S \subseteq \mathbb{Z}$, the Lagrange basis polynomial is defined as $\triangle_{i,S}(z) = \prod_{j \in S, j \neq i} (\frac{z-j}{i-j})$. Let $f(z) \in \mathbb{Z}[z]$ be a d-th degree polynomial. If $|S| = d+1$, from a set of $d+1$ points $\{(i, f(i))\}_{i \in S}$, one can reconstruct $f(z)$ as follows.

$$f(z) = \sum_{i \in S} f(i) \cdot \triangle_{i,S}(z).$$

In our scheme, we will particularly use the interpolation for a first degree polynomial. In particular, let $f(z)$ be a first degree polynomial, one can obtain $f(0)$ from two points $(i_1, f(i_1)), (i_2, f(i_2))$ where $i_1 \neq i_2$ by computing

$$f(0) = f(i_1)\frac{i_2}{i_2 - i_1} + f(i_2)\frac{i_1}{i_1 - i_2}. \tag{1}$$

3 Definitions

3.1 Algorithm Definition

In this section, we provide the syntax definition of Hybrid Revocable Attribute-Based Encryption (HR-ABE). Let $\mathcal{N}$ be the universe of attributes for ABE. Let $\mathcal{A}$ denote a collection of access structures over $\mathcal{N}$ which are allowed to be used in the scheme. Let $\mathcal{T}$, $\mathcal{M}$, $\mathcal{U}$ be the universes of time periods, messages, and user key serial numbers, respectively. A HR-ABE scheme consists of five algorithms:

$\mathsf{Setup}(n) \to (\mathsf{pk}, \mathsf{msk})$**.** This is a randomized algorithm that takes an input n which is the size of $\mathcal{U}$. It outputs the public key pk and a master key msk.

$\mathsf{Encrypt}(\mathsf{mode}, \mathsf{a}, \omega, \mathsf{m}, \mathsf{pk}) \to \mathsf{ct}$**.** This is a randomized algorithm that takes as input $\mathsf{mode} \in \{\text{'dir'}, \text{'ind'}\}$ which representing direct or indirect revocation mode resp., an auxiliary input a being either

$$\mathsf{a} = \begin{cases} \text{a revocation list } R \subseteq \mathcal{U} & \text{if } \mathsf{mode} = \text{'dir'} \\ \text{a present time attribute } t \in \mathcal{T} & \text{if } \mathsf{mode} = \text{'ind'}, \end{cases}$$

a set of attributes $\omega \subseteq \mathcal{N}$, a message $\mathsf{m} \in \mathcal{M}$, and public key pk. It outputs a ciphertext ct.

$\mathsf{KeyGen}(\mathsf{id}, \mathbb{A}, \mathsf{msk}, \mathsf{pk}) \to \mathsf{sk}_{(\mathsf{id},\mathbb{A})}$**.** This is a randomized algorithm that takes as input a serial number $\mathsf{id} \in \mathcal{U}$, an access structure $\mathbb{A} \in \mathcal{A}$, the master key msk, and the public key pk. It outputs a private decryption key $\mathsf{sk}_{(\mathsf{id},\mathbb{A})}$.

$\mathsf{KeyUpdate}(R, t, \mathsf{msk}, \mathsf{pk}) \to \mathsf{uk}_{(R,t)}$**.** This is a randomized algorithm that takes as input a revocation list $R \subseteq \mathcal{U}$, a present time attribute t, the master key msk, and the public key pk. It outputs a key update material $\mathsf{uk}_{(R,t)}$.

$\mathsf{Decrypt}(\mathsf{ct}, (\mathsf{mode}, \mathsf{a}, \omega), \mathsf{sk}_{(\mathsf{id},\mathbb{A})}, (\mathsf{id}, \mathbb{A}), \mathsf{pk}, \mathsf{b}) \to \mathsf{m}$**.** This algorithm takes as input the ciphertext ct that was encrypted under $(\mathsf{mode}, \mathsf{a}, \omega)$, the decryption key $\mathsf{sk}_{(\mathsf{id},\mathbb{A})}$ for user index id with access control structure $\mathbb{A}$, the public key pk, and an auxiliary input b being either

$$\mathsf{b} = \begin{cases} \text{an empty string} & \text{if } \mathsf{mode} = \text{'dir'} \\ \text{an update key } \left(\mathsf{uk}_{(R,t)}, (R, t)\right) & \text{if } \mathsf{mode} = \text{'ind'}. \end{cases}$$

It outputs the message m or a special symbol $\perp$ indicating an unsuccessful decryption.

We require the standard correctness of decryption, that is, if we let $\mathsf{Setup} \to (\mathsf{pk}, \mathsf{msk})$ and $\mathsf{KeyGen}(\mathsf{id}, \mathbb{A}, \mathsf{msk}, \mathsf{pk}) \to \mathsf{sk}_{(\mathsf{id},\mathbb{A})}$ then for all $\mathsf{m} \in \mathcal{M}$; $\mathsf{id} \in \mathcal{U}$; $R \subseteq \mathcal{U}$; $\mathbb{A} \in \mathcal{A}$; $\omega \in \mathcal{N}$, $t \in \mathcal{T}$:

Direct revocation mode. Let $\mathsf{Encrypt}(\text{'dir'}, R, \omega, \mathsf{m}, \mathsf{pk}) \to \mathsf{ct}$. If $\omega \in \mathbb{A}$ and $\mathsf{id} \notin R$, then $\mathsf{Decrypt}\Big(\mathsf{ct}, (\text{'dir'}, R, \omega), \mathsf{sk}_{(\mathsf{id},\mathbb{A})}, (\mathsf{id}, \mathbb{A}), \mathsf{pk}\Big) = \mathsf{m}$.

Indirect revocation mode. Let $\mathsf{Encrypt}(\text{'ind'}, t, \omega, \mathsf{m}, \mathsf{pk}) \to \mathsf{ct}$, $\mathsf{KeyUpdate}(R, t, \mathsf{msk}, \mathsf{pk}) \to \mathsf{uk}_{(R,t)}$. If $\omega \in \mathbb{A}$ and $\mathsf{id} \notin R$, then $\mathsf{Decrypt}\Big(\mathsf{ct}, (\text{'ind'}, t, \omega), \mathsf{sk}_{(\mathsf{id},\mathbb{A})}, (\mathsf{id}, \mathbb{A}), \mathsf{pk}, \mathsf{uk}_{(R,t)}, (R, t)\Big) = \mathsf{m}$.

Augment Definition. In the real usage, as mentioned earlier in Section §1.2, we will wish to assign id as a *unique* serial number for each key generated so far by the key authority. In this case, the authority will maintain the internal state for recording the set $\mathcal{S}$ of assigned id so far. More formally, it will use a simple stateful $\mathsf{KeyGen}^{\mathsf{s}}$ algorithm defined as follows. Initially, $\mathcal{S} = \emptyset$.

$\mathsf{KeyGen}^{\mathsf{s}}(\mathsf{id}, \mathbb{A}, \mathsf{msk}, \mathsf{pk}) \to \mathsf{sk}_{(\mathsf{id},\mathbb{A})}$**.** If $\mathsf{id} \in \mathcal{S}$ then return $\perp$. Else, run $\mathsf{KeyGen}(\mathsf{id}, \mathbb{A}, \mathsf{msk}, \mathsf{pk}) \to \mathsf{sk}_{(\mathsf{id},\mathbb{A})}$. Increment the internal state $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathsf{id}\}$. It then outputs $\mathsf{sk}_{(\mathsf{id},\mathbb{A})}$.

We provide some further discussions on our syntax definition and comparisons to the previous model from [6] in Appendix §A.1.

3.2 Security Notions

We now give the definitions of security notions for HR-ABE. We consider selective-target security, where the adversary is required to specify the target mode, auxiliary input, and attribute set before seeing the public key. Although we consider such a static adversary, the queries can still be asked in an adaptive manner. Indeed, we consider two notions: semi-static and adaptive query model. The semi-static one is similar to the one considered in [13] (in the context of broadcast encryption). Below, we note that the Query phase can be continued after the Challenge phase. We provide the formal definition first then explain later.

Init. The adversary declares the target $(\mathsf{mode}^\star, \mathsf{a}^\star, \omega^\star)$ of encryption mode, its corresponding auxiliary input, and the target attribute set $\omega^\star$. Recall that if $\mathsf{mode}^\star = \text{'dir'}$ then $a^\star = R^\star$ (the target revoked set) and if $\mathsf{mode}^\star = \text{'ind'}$ then $a^\star = t^\star$ (the target present time).

Setup. The challenger runs the Setup of HR-ABE and hands the public key pk to the adversary.

Query Phase. The challenger answers the queries as follows.
If $\mathsf{mode}^\star = \text{'dir'}$, then for each query:

If $\mathcal{A}$ queries for $\mathcal{SK}(\mathsf{id}, \mathbb{A})$ then
  If $\omega^\star \in \mathbb{A}$ then
  If $\mathsf{id} \notin R^\star$ then return $\perp$
 $\mathsf{KeyGen}(\mathsf{id}, \mathbb{A}, \mathsf{msk}, \mathsf{pk}) \to \mathsf{sk}_{(\mathsf{id},\mathbb{A})}$
 Return $\mathsf{sk}_{(\mathsf{id},\mathbb{A})}$
	If $\mathcal{A}$ queries for $\mathcal{UK}(R, t)$ then
	 $\mathsf{KeyUpdate}(R, t, \mathsf{msk}, \mathsf{pk}) \to \mathsf{uk}_{(R,t)}$
	 Return $\mathsf{uk}_{(R,t)}$

If $\mathsf{mode}^\star = \text{'ind'}$, then we consider two variants of attack models.

1. **Semi-static query model.** The adversary first announces $\tilde{R}$. Then, for each query:

If $\mathcal{A}$ queries for $\mathcal{SK}(\mathsf{id}, \mathbb{A})$ then
  If $\omega^\star \in \mathbb{A}$ then
  If $\mathsf{id} \notin \tilde{R}$ then return $\perp$
 $\mathsf{KeyGen}(\mathsf{id}, \mathbb{A}, \mathsf{msk}, \mathsf{pk}) \to \mathsf{sk}_{(\mathsf{id},\mathbb{A})}$
 Return $\mathsf{sk}_{(\mathsf{id},\mathbb{A})}$
	If $\mathcal{A}$ queries for $\mathcal{UK}(R, t)$ then
	  If $t = t^\star$ then
	  If $\tilde{R} \not\subseteq R$ then return $\perp$
	 $\mathsf{KeyUpdate}(R, t, \mathsf{msk}, \mathsf{pk}) \to \mathsf{uk}_{(R,t)}$
	 Return $\mathsf{uk}_{(R,t)}$

2. **Adaptive query model.** The challenger maintains two sets $\mathcal{R}_\mathsf{s}$ and $\mathcal{R}_\mathsf{u}$. Set $\mathcal{R}_\mathsf{s}$ is the set of id for keys corresponding to $\mathbb{A}$ such that $\omega^\star \in \mathbb{A}$. Set $\mathcal{R}_\mathsf{u}$ is the set of id which is revoked via all updated key queries at $t = t^\star$. Initially, $\mathcal{R}_\mathsf{s} = \emptyset$ and $\mathcal{R}_\mathsf{u} = \mathcal{U}$. For each query:

If $\mathcal{A}$ queries for $\mathcal{SK}(\mathsf{id}, \mathbb{A})$ then
  If $\omega^\star \in \mathbb{A}$ then
  If $\mathsf{id} \notin \mathcal{R}_\mathsf{u}$ then return $\perp$
  Else $\mathcal{R}_\mathsf{s} \leftarrow \mathcal{R}_\mathsf{s} \cup \{\mathsf{id}\}$
 $\mathsf{KeyGen}(\mathsf{id}, \mathbb{A}, \mathsf{msk}, \mathsf{pk}) \to \mathsf{sk}_{(\mathsf{id},\mathbb{A})}$
 Return $\mathsf{sk}_{(\mathsf{id},\mathbb{A})}$
	If $\mathcal{A}$ queries for $\mathcal{UK}(R, t)$ then
	  If $t = t^\star$ then
	  If $\mathcal{R}_\mathsf{s} \not\subseteq R$ then return $\perp$
	  Else $\mathcal{R}_\mathsf{u} \leftarrow \mathcal{R}_\mathsf{u} \cap R$
	 $\mathsf{KeyUpdate}(R, t, \mathsf{msk}, \mathsf{pk}) \to \mathsf{uk}_{(R,t)}$
	 Return $\mathsf{uk}_{(R,t)}$

Challenge. The adversary submits two equal length messages m_0 and m_1. The challenger flips a random bit b and computes the challenge ciphertext $\mathsf{ct}^\star$ of m_b on the target $(\mathsf{mode}^\star, \mathsf{a}^\star, \omega^\star)$ and then gives $\mathsf{ct}^\star$ to the adversary.

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

The advantage of an adversary in this game is defined as $\Pr[b = b'] - \frac{1}{2}$. We note that the above definition can be easily extended to handle chosen-ciphertext attacks by allowing decryption queries in the Query Phase.

Definition 3. *An HR-ABE scheme is secure in the sense of indistinguishability against selective-target with semi-static (resp., adaptive) query attack if all polynomial time adversaries have at most a negligible advantage in the above game for respective variants. Denote the two notions by* IND-sHRSS *and* IND-sHRA *respectively.*

Intuition for Security Notion. We explain the intuition behind the above notion. The condition for queries in the direct mode is quite straightforward: the adversary can query secret key only if $\omega^\star \notin \mathbb{A}$ *or* $\mathsf{id} \notin R^\star$ but can query any update key (since it should be useless in this mode).

The condition for the indirect mode is somewhat more complicated. First, consider the adaptive query model, *i.e.,* the query model in IND-sHRA. Basically, if $\mathsf{sk}_{\mathsf{id},\mathbb{A}}$ such that $\omega^\star \in \mathbb{A}$ and $\mathsf{uk}_{R,t^\star}$ such that $\mathsf{id} \notin R$ are known to the adversary, it can be used to decrypt the challenge ciphertext. The important security property we wish to capture is that knowing only one of these two should be useless for any such pairs (*á la* security against collusion). But we never know which one of the two will be queried first, so we capture this by doing the house keeping of both kinds of queries simultaneously and adaptively by using $\mathcal{R}_\mathsf{s}$ and $\mathcal{R}_\mathsf{u}$ defined above. We call a user index id a *critical* user if $\mathcal{SK}(\mathsf{id}, \mathbb{A})$ such that $\omega^\star \in \mathbb{A}$ is asked. $\mathcal{R}_\mathsf{s}$ maintains such a set in this adaptive query model.

Next, we consider the semi-static query model, *i.e.,* the query model in IND-sHRSS. In this notion, the adversary must commit at the beginning of the query phase the set $\tilde{R}$ of id that could potentially be critical. Eventually even if its corresponding $\mathcal{SK}$ query is not asked, the query $\mathcal{UK}(R, t^\star)$ such that $\mathsf{id} \notin R$ is not permitted. Such a $\mathcal{UK}$ query would have been allowed if the adaptive query model were considered. Hence, we conclude that the semi-static model is weaker.

We provide some comparisons to the security model of [6] in Appendix §A.2.

4 A Hybrid Revocable ABE Scheme

4.1 Intuition for Our Scheme

We intuitively explain how our scheme works. First recall the approach of the indirectly revocable IBE and ABE of Boldyreva et al. [6]. It utilizes the binary tree T with n leaves similarly as in the Complete-Subtree method [1,18]. Each user will be associated to a leaf which is unassigned yet. Each node x in the tree is assigned a random secret polynomial f_x of first degree in such a way that

$f_x(0) = \alpha$, where α denotes the master key. The scheme uses $Y^\alpha = e(g,g)^{s\alpha} \in \mathbb{G}_T$ as a message encapsulation key, where s is the randomness in encryption (and we denote $Y = e(g,g)^s$).

To indirectly revoke R at time t, the key authority uses the master key to construct an *atomic* update key corresponding to each node in $\mathsf{Cover}(R)$. Only user who has an ancestor node in $\mathsf{Cover}(R)$, say node x, can compute $Y^{f_x(t)}$. Also, only user whose access structure is satisfied by attribute set associated with ciphertext can compute $Y^{f_x(1)}$. Due to the fact that f_x is first degree, from this two elements, decryption can be done by interpolation in the exponent to yield $Y^{f_x(0)}$.

In our scheme, the indirect mode is done in exactly the same way as above. The rough idea is that we will enable the direct revocation by letting the sender herself generate an atomic update key, albeit at a *dummy time* which is exactly the name of node x for all $x \in \mathsf{Cover}(R)$. Similarly, only user who has an ancestor node in $\mathsf{Cover}(R)$, say node x, can compute $Y^{f_x(x)}$. From this and usual $Y^{f_x(1)}$, he interpolates to obtain $Y^{f_x(0)}$.

The difficulty is how to enable a sender to generate such an atomic update key herself since she is not the key authority and hence cannot construct as in the indirect mode. We solve this by providing a *partial* update key as a part of private key beforehand. This appears as $(D_x^{(3)}, D_x^{(4)})$ in Eq.(2) below, which indeed has the same form as the update key in Eq.(3). This partial key is enabled only when a proper remaining part of update key from sender come with ciphertext.

4.2 The Construction

Some Terminologies. In our scheme, we define the universe set of serial numbers $\mathcal{U}$ as the set of leaves in the complete binary tree $\mathcal{L} = \{1, \ldots, n\}$. Let m be the maximum size of attribute set allowed to be associated with a ciphertext, *i.e.,* we restrict $|\omega| \leq m$. Let d be the maximum of $|\mathsf{Cover}(R)|$ for all $R \subseteq \mathcal{U}$. Our scheme supports any linear secret-sharing access structure which we denote its universe as $\mathcal{A}_{\mathsf{LSSS}}$. Consequently, we let an access structure in its LSSS matrix form (M, π) (*cf.* Definition 2) be input directly to the algorithms in the scheme. We sometimes denote it as $\mathbb{A}_{(M,\pi)}$. Let the attribute space be $\mathcal{N} = \mathbb{Z}_p^*$ and the message space be $\mathcal{M} = \mathbb{G}_T$.

We assume that $\mathcal{T}, \mathcal{X} \subseteq \mathbb{Z}_p^* \backslash \{1\}$ and that $\mathcal{T} \cap \mathcal{X} = \emptyset$. This is wlog since we can extend to the case where domains are $\mathcal{T}' = \mathcal{X}' = \{0,1\}^*$ by using a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p^* \setminus \{1\}$ and then using $H(0||x) \in \mathcal{X}$ instead of $x \in \mathcal{X}'$ and $H(1||t) \in \mathcal{T}$ instead of $t \in \mathcal{T}'$ in the scheme. In this case, the collision resistance ensures that $H(0||x) \neq H(1||t)$ for any x, t, hence $\mathcal{T} \cap \mathcal{X} = \emptyset$. We are now ready to describe our scheme.

▶ $\mathsf{Setup}(n)$: The algorithm first picks a random generator $g \in \mathbb{G}$ and also randomly chooses $u_0, \ldots, u_d, h_0, \ldots, h_m \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$. The public key is: $\mathsf{pk} = \big(g, e(g,g)^\alpha, u_0, \ldots, u_d, h_0, \ldots, h_m\big)$. For all node $x \in \mathcal{X}$ in the tree, it randomly chooses $a_x \in \mathbb{Z}_p$ for defining a first degree polynomial $f_x(z) = a_x z + \alpha$. The master key is $\mathsf{msk} = (\alpha, \{a_x\})$. It outputs $(\mathsf{pk}, \mathsf{msk})$. Define a function $P : \mathbb{Z}_p \to \mathbb{G}$ by $P(x) = \prod_{j=0}^{d} u_j^{(x^j)}$. Define a function $F : \mathbb{Z}_p \to \mathbb{G}$ by $F(x) = \prod_{j=0}^{m} h_j^{(x^j)}$.

▶ $\mathsf{Encrypt}(\mathsf{mode}, \mathsf{a}, \omega, \mathsf{m}, \mathsf{pk})$: The algorithm first picks a random $s \in \mathbb{Z}_p$. It then computes

$$C = \mathsf{m} \cdot (e(g,g)^{\alpha})^s, \qquad C^{(1)} = g^s, \qquad C_k^{(2)} = F(k)^s,$$

where the last term is for each $k \in \omega$. From this point, two encryption modes diverge as follows.

Direct revocation mode (mode = 'dir'). In this case, we have $\mathsf{a} = R$, the revocation list, as an input. It runs $\mathsf{Cover}(R)$ to find a minimal node set that covers $\mathcal{U} \setminus R$. It then computes for each $x \in \mathsf{Cover}(R)$: $C_x^{(3)} = P(x)^s$. It outputs the ciphertext as $\mathsf{ct} = \big(C, C^{(1)}, \{C_k^{(2)}\}_{k\in\omega}, \{C_x^{(3)}\}_{x\in\mathsf{Cover}(R)}\big)$.

Indirect revocation mode (mode = 'ind'). In this case, we have $\mathsf{a} = t$, the present time attribute, as an input. It computes $C^{(3)} = P(t)^s$. It then outputs the ciphertext as $\mathsf{ct} = \big(C, C^{(1)}, \{C_k^{(2)}\}_{k\in\omega}, C^{(3)}\big)$.

▶ $\mathsf{KeyGen}(\mathsf{id}, (M,\pi), \mathsf{msk}, \mathsf{pk})$: The input consists of a serial number $\mathsf{id} \in \mathcal{U}$ (which is a leaf in the binary tree), a LSSS access structure $(M,\pi) \in \mathcal{A}_{\mathsf{LSSS}}$, the master key and the public key. Let M be $\ell \times k$ matrix. The algorithm computes a key as follows.

For all $x \in \mathsf{Path}(\mathsf{id})$, it first shares $f_x(1)$ with the LSSS (M,π) as follows. It chooses $z_{x,2} \ldots, z_{x,k} \in \mathbb{Z}_p$ and lets $\boldsymbol{v}_x = (f_x(1), z_{x,2}, \ldots, z_{x,k})$. For $i = 1$ to ℓ, it calculates the share $\lambda_{x,i} = \boldsymbol{M_i} \cdot \boldsymbol{v}_x$, where $\boldsymbol{M_i}$ is the vector corresponding to ith row of M. It then randomly chooses $r_{x,1}, \ldots, r_{x,\ell}, r_x \in \mathbb{Z}_p$. It outputs the private key as $\mathsf{sk}_{(\mathsf{id},(M,\pi))} = \big((D_{x,i}^{(1)}, D_{x,i}^{(2)})_{x\in\mathsf{Path}(\mathsf{id}),i\in[1,\ell]}, (D_x^{(3)}, D_x^{(4)})_{x\in\mathsf{Path}(\mathsf{id})}\big)$ where

$$\begin{aligned} D_{x,i}^{(1)} &= g^{\lambda_{x,i}} F(\pi(i))^{r_{x,i}}, & D_{x,i}^{(2)} &= g^{r_{x,i}}. \\ D_x^{(3)} &= g^{f_x(x)} P(x)^{r_x}, & D_x^{(4)} &= g^{r_x}. \end{aligned} \tag{2}$$

▶ $\mathsf{KeyUpdate}(R, t, \mathsf{msk}, \mathsf{pk})$: The algorithm first runs $\mathsf{Cover}(R)$ to find a minimal node set that covers $\mathcal{U} \setminus R$. For each $x \in \mathsf{Cover}(R)$, it randomly chooses $r_x \in \mathbb{Z}_p$ and sets the key update material as $\mathsf{uk}_{(R,t)} = (U_x^{(1)}, U_x^{(2)})_{x\in\mathsf{Cover}(R)}$ where

$$U_x^{(1)} = g^{f_x(t)} P(t)^{r_x}, \qquad U_x^{(2)} = g^{r_x}. \tag{3}$$

▶ $\mathsf{Decrypt}(\mathsf{ct}, (\mathsf{mode}, \mathsf{a}, \omega), \mathsf{sk}_{(\mathsf{id},\mathbb{A})}, (\mathsf{id}, \mathbb{A}), \mathsf{pk}, \mathsf{b})$: Suppose that ω satisfies (M,π) and $\mathsf{id} \notin R$. (So that the decryption is possible). Let $I = \{i |\ \pi(i) \in \omega\}$. It then calculates the corresponding set of reconstruction constants $\{(i,\nu_i)\}_{i\in I} = \mathsf{Recon}_{(M,\pi)}(\omega)$. Since $\mathsf{id} \notin R$, it also finds a node x such that $x \in \mathsf{Path}(\mathsf{id}) \cap \mathsf{Cover}(R)$. Then it computes the following

$$K' = \prod_{i=1}^{\ell} \left(\frac{e(D_{x,i}^{(1)}, C^{(1)})}{e(C_{\pi(i)}^{(2)}, D_{x,i}^{(2)})} \right)^{\nu_i}.$$

From this point, two modes diverges as follows.

$$K = \begin{cases} (K')^{\frac{x}{x-1}} \cdot \left(\frac{e(D_x^{(3)}, C^{(1)})}{e(C_x^{(3)}, D_x^{(4)})}\right)^{\frac{1}{1-x}} & \text{if mode} = \text{'dir'} \\ (K')^{\frac{t}{t-1}} \cdot \left(\frac{e(U_x^{(1)}, C^{(1)})}{e(C^{(3)}, U_x^{(2)})}\right)^{\frac{1}{1-t}} & \text{if mode} = \text{'ind'} \end{cases}$$

Finally, it obtains message $\mathsf{m} = C/K$.

Correctness. We first claim that $K' = e(g,g)^{sf_x(1)}$, which can be verified as follows.

$$K' = \prod_{i=1}^{\ell} \left(\frac{e(g^{\lambda_{x,i}} F(\pi(i))^{r_{x,i}}, g^s)}{e(F(\pi(i))^s, g^{r_{x,i}})}\right)^{\nu_i} = \prod_{i=1}^{\ell} e(g,g)^{s\lambda_{x,i}\nu_i} = e(g,g)^{sf_x(1)},$$

where the first equality is due to the construction, the second one is from the property of bilinear map, and the third one is due to the linear reconstruction property of linear secret sharing scheme.

Now for direct revocation mode, we can verify its correctness as follows.

$$\begin{aligned} K &= (K')^{\frac{x}{x-1}} \cdot \left(\frac{e(g^{f_x(x)} P(x)^{r_x}, g^s)}{e(P(x)^s, g^{r_x})}\right)^{\frac{1}{1-x}} \\ &= (e(g,g)^{sf_x(1)})^{\frac{x}{x-1}} \cdot (e(g,g)^{sf_x(x)})^{\frac{1}{1-x}} = e(g,g)^{s\alpha}, \end{aligned}$$

which is indeed the Lagrange interpolation from two points $(1, f_x(1)), (x, f_x(x))$ to evaluate $f_x(0) = \alpha$ (in the exponent) as in Eq.(1).

Similarly, for indirect mode, we can verify its correctness as follows.

$$\begin{aligned} K &= (K')^{\frac{t}{t-1}} \cdot \left(\frac{e(g^{f_x(t)} P(t)^{r_x}, g^s)}{e(P(t)^s, g^{r_x})}\right)^{\frac{1}{1-t}} \\ &= (e(g,g)^{sf_x(1)})^{\frac{t}{t-1}} \cdot (e(g,g)^{sf_x(t)})^{\frac{1}{1-t}} = e(g,g)^{s\alpha}, \end{aligned}$$

which is indeed the Lagrange interpolation from two points $(1, f_x(1)), (t, f_x(t))$ to evaluate $f_x(0) = \alpha$ (in the exponent) as in Eq.(1).

Theorem 1. *If an adversary can break the above scheme with advantage ϵ in the sense of* IND-sHRSS*, then a simulator with advantage ϵ in solving the DBDH problem can be constructed.*

Discussion on the Revocable ABE of [6]. The indirectly revocable ABE of Boldyreva et al. [6] can be derived from our scheme by just neglecting the direct mode and deleting the term $(D_x^{(3)}, D_x^{(4)})$ from the private key. Note that such a scheme was not explicitly given in their paper though. We hence try to prove the security of their implicit scheme by ourselves in the adaptive query model. We found that by extending their proof from the IBE case to the ABE case, the security reduction is lost by multiplicative factor of $O(1/2^q)$, where q is the

number of queries to either secret key or update key oracle. Intuitively, this is due to the fact that the simulator in the proof has to guess whether a user will be critical user or revoked user at the target time $t^\star$ (see discussion in Section §3.2), in order to simulate answers to queries. If only one guess fails, it will abort. The simulation will thus succeed with only probability $1/2^q$. Since q is polynomial in security parameter, this lost is exponential. Therefore, the security in the adaptive query model would not hold for their scheme. We conclude here that their revocable ABE is proved secure only in the semi-static query model. Note that in their IBE case, the reduction cost is only $1/2$ since the simulator only needs to guess once for the only one critical user candidate, which is the challenge identity (denoted by ω^* in their paper).

Some Extensions. Note that n is fixed in our main scheme but can be extended as suggested in [6] by adding a new root over the old one, hence the universe becomes of size $2n$. The new key at the new root is distributed via key update. The chosen-ciphertext secure scheme can also be obtained similarly as in [6].

5 Security Proof

In this section, we give the security proof of our scheme as stated in Theorem 1.

Proof. Suppose there exists an adversary, $\mathcal{A}$, that has advantage ϵ in attacking the proposed scheme. We build a simulator $\mathcal{B}$ that solves the Decision BDH problem in $\mathbb{G}$. $\mathcal{B}$ is given as input a random DBDH challenge $\boldsymbol{Y} = (g, g^a, g^b, g^s, Z)$, where Z is either $e(g,g)^{abs}$ or a random element in $\mathbb{G}_1$. $\mathcal{B}$ proceeds as follows.

Init. The selective-target game begins with $\mathcal{A}$ first outputting a target ($\mathsf{mode}^\star$, $\mathsf{a}^\star, \omega^\star$) that it intends to attack. The proof for two cases of $\mathsf{mode}^\star$ diverges here.

5.1 When Target Mode Is Direct Revocation

We first consider the case where $\mathsf{mode}^\star$ = 'dir'. In this case, we have the auxiliary input $\mathsf{a}^\star = R^\star$.

Setup. To generate pk, algorithm $\mathcal{B}$ first defines

$$q(x) = x^{m-|\omega^\star|} \cdot \prod_{k \in \omega^\star} (x - k) = \sum_{j=0}^{m} q_j x^j.$$

We note that q_j's terms can be computed completely from $\omega^\star$. From this we can ensure that $q(x) = 0$ if and only if $x \in \omega^\star$. It then randomly picks a degree m polynomial in $\mathbb{Z}_p[x]$ as $\phi(x) = \sum_{j=0}^{m} \phi_j x^j$. It lets $h_j = (g^a)^{q_j} g^{\phi_j}$ for $j = 0, \ldots, m$. We thus have $F(x) = \prod_{j=0}^{m} h_j^{(x^j)} = (g^a)^{q(x)} \cdot g^{\phi(x)}$.

Similarly, it also defines

$$p(y) = y^{d-|\mathsf{Cover}(R^\star)|} \cdot \prod_{x \in \mathsf{Cover}(R^\star)} (y - x) = \sum_{j=0}^{d} p_j y^j.$$

From this we can ensure that for all $x \in \mathcal{X}$, $p(x) = 0$ if and only if $x \in \mathsf{Cover}(R^\star)$ and that for all $t \in \mathcal{T}$, $p(t) \neq 0$. The latter is since $\mathcal{X} \cap \mathcal{T} = \emptyset$.

It then randomly picks a degree d polynomial in $\mathbb{Z}_p[x]$ as $\rho(y) = \sum_{j=0}^{d} \rho_j y^j$. It lets $u_j = (g^a)^{p_j} g^{\rho_j}$ for $j = 0, \ldots, d$. We thus have $P(y) = \prod_{j=0}^{d} u_j^{(y^j)} = (g^a)^{p(y)} \cdot g^{\rho(y)}$.

The algorithm $\mathcal{B}$ implicitly defines $\alpha = ab$. It gives $\mathcal{A}$ the public key $\mathsf{pk} = (g, e(g^a, g^b), u_0, \ldots, u_d, h_0, \ldots, h_m)$. Since g, a, b, ϕ, ρ are chosen randomly and independently, pk has an identical distribution to that in the actual construction.

Let $\mathcal{X}_{R^\star} = \{x \in \mathsf{Path}(\mathsf{id}) \mid \mathsf{id} \in R^\star\}$. For all node x in the binary tree, it randomly chooses $a'_x \in \mathbb{Z}_p$ and implicitly pre-defines:

$$a_x = \begin{cases} a'_x - ab & \text{if } x \in \mathcal{X}_{R^\star} \\ a'_x - \frac{ab}{x} & \text{if } x \notin \mathcal{X}_{R^\star}. \end{cases} \tag{4}$$

Note that the simulator cannot compute these values, since ab is unknown. Intuitively, predefining is done so that we can compute

$$f_x(1) = a_x(1) + ab = (a'_x - ab) + ab = a'_x \qquad \text{if } x \in \mathcal{X}_{R^\star} \tag{5}$$

$$f_x(x) = a_x(x) + ab = (a'_x - \frac{ab}{x})x + ab = a'_x x \qquad \text{if } x \notin \mathcal{X}_{R^\star} \tag{6}$$

Query Phase. The adversary makes requests for private keys corresponding to user index and access structure pair $(\mathsf{id}, (M, \pi))$ subjected to condition that $\omega^\star$ does not satisfy (M, π) *or* $\mathsf{id} \in R^\star$. As a big picture, the algorithm $\mathcal{B}$ simulates answers as follows.

If $\mathcal{A}$ queries for $\mathcal{SK}(\mathsf{id}, \mathbb{A}_{(M,\pi)})$ then
  If $\omega^\star \in \mathbb{A}_{(M,\pi)}$ then
  If $\mathsf{id} \notin R^\star$ then return $\perp$
  Else do **Case S1**
  If $\omega^\star \notin \mathbb{A}_{(M,\pi)}$ then
  Do **Case S2**
	If $\mathcal{A}$ queries for $\mathcal{UK}(R, t)$ then
	  Do **Case U**

[**Case S1:** $\omega^\star \in \mathbb{A}_{(M,\pi)}$ and $\mathsf{id} \in R^\star$] First we claim that in this case we can compute $f_x(1)$ for all $x \in \mathsf{Path}(\mathsf{id})$. This is since $\mathsf{id} \in R^\star$, hence for all $x \in \mathsf{Path}(\mathsf{id})$ we have $x \in \mathcal{X}_{R^\star}$. Hence $f_x(1)$ can be computed by Eq.(5).

To create $(D^{(1)}_{x,i}, D^{(2)}_{x,i})_{i \in [1,\ell], x \in \mathsf{Path}(\mathsf{id})}$, algorithm $\mathcal{B}$ just computes as in the real construction since it knows $f_x(1)$.

To create $(D^{(3)}_x, D^{(4)}_x)_{x \in \mathsf{Path}(\mathsf{id})}$, it does as follows. For all $x \in \mathsf{Path}(\mathsf{id})$, it randomly chooses $r'_x \in \mathbb{Z}_p$ and computes

$$D^{(3)}_x = (g^{a'_x})^x (g^b)^{\left(-\frac{(1-x)\rho(x)}{p(x)}\right)} P(x)^{r'_x}, \qquad D^{(4)}_x = (g^b)^{-\frac{(1-x)}{p(x)}} g^{r'_x}.$$

Note that these can be computed since, in particular, $p(x) \neq 0$ due to the fact that for all $x \in \mathcal{X}$, $p(x) = 0$ iff $x \in \mathsf{Cover}(R^\star)$ but here since $x \in \mathcal{X}_{R^\star}$ we

have $x \notin \mathsf{Cover}(R^\star)$. We claim that $(D_x^{(3)}, D_x^{(4)})$ is valid by implicitly letting $r_x = r'_x - \frac{b(1-x)}{p(x)}$ and seeing that

$$\begin{aligned} D_x^{(3)} &= (g^{a'_x})^x (g^b)^{\left(-\frac{(1-x)\rho(x)}{p(x)}\right)} P(x)^{r'_x} \\ &= (g^{a'_x x + (1-x)ab})(g^{ab})^{-(1-x)} (g^b)^{\left(-\frac{(1-x)\rho(x)}{p(x)}\right)} P(x)^{r'_x} \\ &= (g^{f_x(x)})((g^a)^{p(x)} g^{\rho(x)})^{\left(-\frac{b(1-x)}{p(x)}\right)} P(x)^{r'_x} = g^{f_x(x)} P(x)^{r_x}. \end{aligned}$$

The validity of $D_x^{(4)}$ is immediate.

[**Case S2:** $\omega^\star \notin \mathbb{A}_{(M,\pi)}$] To create $\left((D_{x,i}^{(1)}, D_{x,i}^{(2)})_{i\in[1,\ell]}, (D_x^{(3)}, D_x^{(4)})\right)_{x\in\mathsf{Path(id)}}$, we categorize node $x \in \mathsf{Path(id)}$ whether it is in $\mathcal{X}_{R^\star}$ or not as follows.

Category 1: $x \in \mathsf{Path(id)} \cap \mathcal{X}_{R^\star}$. The simulator $\mathcal{B}$ can create the key in exactly the same manner as in the case **S1**, since in this category $x \in \mathcal{X}_{R^\star}$.

Category 2: $x \in \mathsf{Path(id)} \setminus \mathcal{X}_{R^\star}$. The simulator $\mathcal{B}$ does as follows. Recall that $\omega^\star \notin \mathbb{A}_{(M,\pi)}$. Hence from Lemma 1, there must exist a vector $\boldsymbol{w} = (w_1, \ldots, w_k) \in \mathbb{Z}_p^k$ such that $w_1 = 1$ and that for all i where $\pi(i) \in \omega^\star$, it holds that $\boldsymbol{M_i} \cdot \boldsymbol{w} = 0$.

To create $(D_{x,i}^{(1)}, D_{x,i}^{(2)})_{i\in[1,\ell]}$, it randomly chooses $z'_{x,2}, \ldots, z'_{x,k} \in \mathbb{Z}_p$ and lets $\boldsymbol{v}'_x = (0, z'_{x,2}, \ldots, z'_{x,k})$. It implicitly defines a vector

$$\boldsymbol{v}_x = (a_x + \alpha)\boldsymbol{w} + \boldsymbol{v}'_x \overset{(4)}{=} \left(a'_x + \alpha(1 - \frac{1}{x})\right)\boldsymbol{w} + \boldsymbol{v}'_x,$$

which will be used for creating shares of $f_x(1) = a_x + \alpha$ as in the construction. For simplicity, let $A = 1 - \frac{1}{x}$.

For i where $\pi(i) \in \omega^\star$, it randomly chooses $r_{x,i} \in \mathbb{Z}_p$ and computes

$$D_{x,i}^{(1)} = g^{\boldsymbol{M_i}\cdot\boldsymbol{v}'_x} F(\pi(i))^{r_{x,i}} = g^{\boldsymbol{M_i}\cdot\boldsymbol{v}_x} F(\pi(i))^{r_{x,i}}, \quad D_{x,i}^{(2)} = g^{r_{x,i}} \tag{7}$$

where the right-hand equality of $D_i^{(1)}$ is due to $\boldsymbol{M_i} \cdot \boldsymbol{w} = 0$.

For i where $\pi(i) \notin \omega^\star$, it randomly chooses $r'_{x,i} \in \mathbb{Z}_p$. It then computes

$$\begin{aligned} D_{x,i}^{(1)} &= (g^{a'_x})^{\boldsymbol{M_i}\cdot\boldsymbol{w}} g^{\boldsymbol{M_i}\cdot\boldsymbol{v}'_x} (g^b)^{\left(-\frac{A(\boldsymbol{M_i}\cdot\boldsymbol{w})\phi(\pi(i))}{q(\pi(i))}\right)} F(\pi(i))^{r'_{x,i}}, \\ D_{x,i}^{(2)} &= (g^b)^{-\frac{A(\boldsymbol{M_i}\cdot\boldsymbol{w})}{q(\pi(i))}} g^{r'_{x,i}}. \end{aligned} \tag{8}$$

Note that these can be computed since, in particular, $q(\pi(i)) \neq 0$ due to the fact that $q(x) = 0$ iff $x \in \omega^\star$ and here $\pi(i) \notin \omega^\star$. We claim that $(D_{x,i}^{(1)}, D_{x,i}^{(2)})$ is valid by implicitly letting $r_{x,i} = r'_{x,i} - b\frac{A(\boldsymbol{M_i}\cdot\boldsymbol{w})}{q(\pi(i))}$ and seeing that

$$\begin{aligned} D_{x,i}^{(1)} &= (g^{a'_x})^{\boldsymbol{M_i}\cdot\boldsymbol{w}} g^{\boldsymbol{M_i}\cdot\boldsymbol{v}'_x} (g^b)^{\left(-\frac{A(\boldsymbol{M_i}\cdot\boldsymbol{w})\phi(\pi(i))}{q(\pi(i))}\right)} F(\pi(i))^{r'_{x,i}} \\ &= (g^{a'_x + abA})^{\boldsymbol{M_i}\cdot\boldsymbol{w}} g^{\boldsymbol{M_i}\cdot\boldsymbol{v}'_x} (g^{ab})^{-A(\boldsymbol{M_i}\cdot\boldsymbol{w})} (g^b)^{\left(-\frac{A(\boldsymbol{M_i}\cdot\boldsymbol{w})\phi(\pi(i))}{q(\pi(i))}\right)} F(\pi(i))^{r'_{x,i}} \\ &= g^{\boldsymbol{M_i}\cdot\boldsymbol{v}_x} ((g^a)^{q(\pi(i))} g^{\phi(\pi(i))})^{-\frac{bA(\boldsymbol{M_i}\cdot\boldsymbol{w})}{q(\pi(i))}} F(\pi(i))^{r'_{x,i}} = g^{\boldsymbol{M_i}\cdot\boldsymbol{v}_x} F(\pi(i))^{r_{x,i}}, \end{aligned}$$

and the term $D_{x,i}^{(2)}$ is immediate.

To create $(D_x^{(3)}, D_x^{(4)})$, it randomly chooses $r_x \in \mathbb{Z}_p$ and computes

$$D_x^{(3)} = (g^{a'_x x})P(x)^{r_x}, \qquad D_x^{(4)} = g^{r_x},$$

which is valid since $f_x(x) = a'_x x$ due to Eq.(6).

[**Case U**] For any R, t, it computes $\mathsf{uk}_{(R,t)} = (U_x^{(1)}, U_x^{(2)})_{x \in \mathsf{Cover}(R)}$ as follows. It first randomly chooses $r'_x \in \mathbb{Z}_p$ and computes

$$\text{If } x \in \mathcal{X}_{R^\star} \begin{cases} U_x^{(1)} &= (g^{a'_x})^t (g^b)^{\left(-\frac{(1-t)\rho(t)}{p(t)}\right)} P(t)^{r'_x}, \\ U_x^{(2)} &= (g^b)^{-\frac{(1-t)}{p(t)}} g^{r'_x}. \end{cases}$$

$$\text{If } x \notin \mathcal{X}_{R^\star} \begin{cases} U_x^{(1)} &= (g^{a'_x})^t (g^b)^{\left(-\frac{(x-t)\rho(t)}{xp(t)}\right)} P(t)^{r'_x}, \\ U_x^{(2)} &= (g^b)^{-\frac{(x-t)}{xp(t)}} g^{r'_x}. \end{cases}$$

Note that these can be computed since, in particular, $p(t) \neq 0$ for any $t \in \mathcal{T}$.

The term $(U_x^{(1)}, U_x^{(2)})$ can be proved valid with implicit randomness $r_x = r'_x - \frac{b(1-t)}{p(t)}$ for the case $x \in \mathcal{X}_{R^\star}$ and $r_x = r'_x - \frac{(x-t)}{xp(t)}$ for the case $x \notin \mathcal{X}_{R^\star}$.

Challenge. The adversary gives two message M_0, M_1 to the simulator. The simulator flips a coin b and creates $C = M_b \cdot Z$, $C^{(1)} = g^s$, for $k \in \omega^\star$, $C_k^{(2)} = (g^s)^{\phi(k)}$, and for $x \in \mathsf{Cover}(R^\star)$, $C_x^{(3)} = (g^s)^{\rho(x)}$. We claim that if $Z = e(g,g)^{abs}$, then the above ciphertext is a valid challenge. The term $C, C^{(1)}$ is trivial. For all $k \in \omega^\star$, we have $q(k) = 0$, hence

$$C_k^{(2)} = (g^s)^{\phi(k)} = ((g^a)^{q(k)} g^{\phi(k)})^s = F(k)^s.$$

Also, for $x \in \mathsf{Cover}(R^\star)$ we have $p(x) = 0$, hence

$$C_x^{(3)} = (g^s)^{\rho(x)} = ((g^a)^{p(x)} g^{\rho(x)})^s = P(x)^s,$$

which concludes our claim.

5.2 When Target Mode Is Indirect Revocation

We now consider the case where $\mathsf{mode}^\star =$ 'ind'. In this case, we have the auxiliary input $\mathsf{a}^\star = t^\star$.

Setup. To generate pk, algorithm $\mathcal{B}$ first defines $q(x), \phi(x)$ and corresponding h_j's as in the previous mode. Similarly, it also defines

$$p(y) = y^{d-1} \cdot (y - t^\star) = \sum_{j=0}^{d} p_j y^j.$$

From this we can ensure that for all $t \in \mathcal{T}$, $p(t) = 0$ if and only if $t = t^\star$ and that for $x \in \mathcal{X}$, $p(x) \neq 0$. The latter is due to $\mathcal{T} \cap \mathcal{X} = \emptyset$.

It then randomly picks a degree d polynomial in $\mathbb{Z}_p[x]$ as $\rho(y) = \sum_{j=0}^{d} \rho_j y^j$. It lets $u_j = (g^a)^{p_j} g^{\rho_j}$ for $j = 0, \ldots, d$. We thus have $P(y) = \prod_{j=0}^{d} u_j^{(y^j)} = (g^a)^{p(y)} \cdot g^{\rho(y)}$.

The algorithm $\mathcal{B}$ implicitly defines $\alpha = ab$. It gives $\mathcal{A}$ the public key $\mathsf{pk} = (g, e(g^a, g^b), u_0, \ldots, u_d, h_0, \ldots, h_m)$. Since g, a, b, ϕ, ρ are chosen randomly and independently, pk has an identical distribution to that in the actual construction.

Query Phase. Since we deal with the semi-static query notion, at the beginning of this phase the adversary $\mathcal{A}$ announces $\tilde{R}$. Let $\mathcal{X}_{\tilde{R}} = \{x \in \mathsf{Path}(\mathsf{id}) \mid \mathsf{id} \in \tilde{R}\}$. For all node x in the binary tree, it randomly chooses $a'_x \in \mathbb{Z}_p$ and implicitly pre-defines:

$$a_x = \begin{cases} a'_x - ab & \text{if } x \in \mathcal{X}_{\tilde{R}} \\ a'_x - \frac{ab}{t^\star} & \text{if } x \notin \mathcal{X}_{\tilde{R}}. \end{cases} \tag{9}$$

Note that the simulator cannot compute these values, since ab is unknown. Intuitively, predefining is done so that we can compute

$$f_x(1) = a_x(1) + ab = (a'_x - ab) + ab = a'_x \qquad \text{if } x \in \mathcal{X}_{\tilde{R}} \tag{10}$$

$$f_x(t^\star) = a_x(t^\star) + ab = (a'_x - \frac{ab}{t^\star})t^\star + ab = a'_x t^\star \qquad \text{if } x \notin \mathcal{X}_{\tilde{R}} \tag{11}$$

The algorithm $\mathcal{B}$ then simulates answers to queries as follows.

If $\mathcal{A}$ queries for $\mathcal{SK}(\mathsf{id}, \mathbb{A})$ then	If $\mathcal{A}$ queries for $\mathcal{UK}(R, t)$ then
If $\omega^\star \in \mathbb{A}_{(M,\pi)}$ then	If $t = t^\star$ then
If $\mathsf{id} \notin \tilde{R}$ then return $\perp$	If $\tilde{R} \not\subseteq R$ then return $\perp$
Else do **Case S1′**	Else do **Case U1′**
If $\omega^\star \notin \mathbb{A}_{(M,\pi)}$ then	If $t \neq t^\star$ then
Do **Case S2′**	Do **Case U2′**

[**Case S1′**: $\omega^\star \in \mathbb{A}_{(M,\pi)}$ and $\mathsf{id} \in \tilde{R}$] The algorithm $\mathcal{B}$ computes $(D^{(1)}_{x,i}, D^{(2)}_{x,i})_{i\in[1,\ell]}$, $D^{(3)}_x, D^{(4)}_x$ for all $x \in \mathsf{Path}(\mathsf{id})$ in exactly the same manner as in the case **S1** albeit using set $\tilde{R}$ instead of $R^\star$ and using polynomial p, ρ defined in Setup of this mode above.

To see why this is valid, notice that in this case for all $x \in \mathsf{Path}(\mathsf{id})$ we have $x \in \mathcal{X}_{\tilde{R}}$, where in such a case we define $a_x = a'_x - ab$ in Eq.(9), which is unchanged from the previous mode.

[**Case S2′**: $\omega^\star \notin \mathbb{A}_{(M,\pi)}$] To create $\big((D^{(1)}_{x,i}, D^{(2)}_{x,i})_{i\in[1,\ell]}, (D^{(3)}_x, D^{(4)}_x)\big)_{x\in\mathsf{Path}(\mathsf{id})}$, we categorize node $x \in \mathsf{Path}(\mathsf{id})$ whether it is in $\mathcal{X}_{\tilde{R}}$ or not as follows.

Category 1: $x \in \mathsf{Path}(\mathsf{id}) \cap \mathcal{X}_{\tilde{R}}$. The simulator $\mathcal{B}$ can create the key in exactly the same manner as in the case **S1′**, since in this category $x \in \mathcal{X}_{\tilde{R}}$.

Category 2: $x \in \mathsf{Path}(\mathsf{id}) \setminus \mathcal{X}_{\tilde{R}}$. The simulator $\mathcal{B}$ does as follows. Recall that $\omega^\star \notin \mathbb{A}_{(M,\pi)}$. Hence from Lemma 1, there must exist a vector $\boldsymbol{w} = (w_1, \ldots, w_k) \in \mathbb{Z}_p^k$ such that $w_1 = 1$ and that for all i where $\pi(i) \in \omega^\star$, it holds that $\boldsymbol{M_i} \cdot \boldsymbol{w} = 0$.

To create $(D^{(1)}_{x,i}, D^{(2)}_{x,i})_{i\in[1,\ell]}$, similarly to the case **S2**, the simulator $\mathcal{B}$ does as follows. It randomly chooses $z'_{x,2}, \ldots, z'_{x,k} \in \mathbb{Z}_p$ and lets $\boldsymbol{v}'_x = (0, z'_{x,2}, \ldots, z'_{x,k})$. It implicitly defines a vector

$$\boldsymbol{v}_x = (a_x + \alpha)\boldsymbol{w} + \boldsymbol{v}'_x \overset{(9)}{=} \big(a'_x + \alpha(1 - \frac{1}{t^\star})\big)\boldsymbol{w} + \boldsymbol{v}'_x,$$

which will be used for creating shares of $f_x(1) = a_x + \alpha$ as in the construction. For simplicity, let $A = 1 - \frac{1}{t^\star}$. The computation of $(D^{(1)}_{x,i}, D^{(2)}_{x,i})_{i\in[1,\ell]}$ can be done using Eq.(7) and Eq.(8) as in the case **S2**, here the only difference is the value of A. The validity thus can be verified similarly.

To create $(D^{(3)}_x, D^{(4)}_x)$, it randomly chooses $r_x \in \mathbb{Z}_p$ and computes

$$D^{(3)}_x = (g^{a'_x})^x (g^b)^{\left(-\frac{(t^\star - x)\rho(x)}{t^\star p(x)}\right)} P(x)^{r'_x}, \qquad D^{(4)}_x = (g^b)^{-\frac{(t^\star - x)}{t^\star p(x)}} g^{r'_x},$$

Note that these can be computed since, in particular, $p(x) \neq 0$ for any $x \in \mathcal{X}$. This can be proved valid with the implicit randomness $r_x = r'_x - \frac{b(t^\star - x)}{t^\star p(x)}$.

[**Case U1′**: $t = t^\star$ and $\tilde{R} \subseteq R$.] To create $(U^{(1)}_x, U^{(2)}_x)_{x\in\mathsf{Cover}(R)}$, $\mathcal{B}$ randomly chooses $r_x \in \mathbb{Z}_p$ for each $x \in \mathsf{Cover}(R)$ and computes

$$U^{(1)}_x = (g^{a'_x t^\star}) P(t^\star)^{r_x}, \qquad U^{(2)}_x = g^{r_x}.$$

We prove that this is valid as follows. Since $\tilde{R} \subseteq R$, hence for all $x \in \mathsf{Cover}(R)$ we have $x \notin \mathcal{X}_{\tilde{R}}$. Therefore from Eq.(11), we have $f_x(t^\star) = a'_x t^\star$.

[**Case U2′**: $t \neq t^\star$.] To create $(U^{(1)}_x, U^{(2)}_x)_{x\in\mathsf{Cover}(R)}$, $\mathcal{B}$ randomly chooses $r'_x \in \mathbb{Z}_p$ for each $x \in \mathsf{Cover}(R)$ and computes

$$\text{If } x \in \mathsf{Cover}(R) \cap \mathcal{X}_{\tilde{R}} \begin{cases} U^{(1)}_x &= (g^{a'_x})^t (g^b)^{\left(-\frac{(1-t)\rho(t)}{p(t)}\right)} P(t)^{r'_x}, \\ U^{(2)}_x &= (g^b)^{-\frac{(1-t)}{p(t)}} g^{r'_x}. \end{cases}$$

$$\text{If } x \in \mathsf{Cover}(R) \setminus \mathcal{X}_{\tilde{R}} \begin{cases} U^{(1)}_x &= (g^{a'_x})^t (g^b)^{\left(-\frac{(t^\star - t)\rho(t)}{t^\star p(t)}\right)} P(t)^{r'_x}, \\ U^{(2)}_x &= (g^b)^{-\frac{(t^\star - t)}{t^\star p(t)}} g^{r'_x}. \end{cases}$$

Note that these can be computed since, in particular, $p(t) \neq 0$ due to the fact that for $t \in \mathcal{T}$, $p(t) = 0$ iff $t = t^\star$.

The term $(U^{(1)}_x, U^{(2)}_x)$ can be proved valid with implicit randomness $r_x = r'_x - \frac{b(1-t)}{p(t)}$ for the case $x \in \mathcal{X}_{R^\star}$ and $r_x = r'_x - \frac{(t^\star - t)}{t^\star p(t)}$ for the case $x \notin \mathcal{X}_{R^\star}$.

Challenge. The adversary gives two message M_0, M_1 to the simulator. The simulator flips a coin b and creates $C = M_b \cdot Z$, $C^{(1)} = g^s$, for $k \in \omega^\star$, $C^{(2)}_k = (g^s)^{\phi(k)}$, and $C^{(3)} = (g^s)^{\rho(t^\star)}$. We claim that if $Z = e(g,g)^{abs}$, then the above ciphertext is a valid challenge. The term $C, C^{(1)}$ is trivial. For all $k \in \omega^\star$, we have $q(k) = 0$, hence

$$C^{(2)}_k = (g^s)^{\phi(k)} = ((g^a)^{q(k)} g^{\phi(k)})^s = F(k)^s.$$

Also, since $p(t^\star) = 0$, hence

$$C^{(3)} = (g^s)^{\rho(t^\star)} = ((g^a)^{p(t^\star)} g^{\rho(t^\star)})^s = P(t^\star)^s,$$

which concludes our claim.

5.3 The Rest of the Proof

Guess. In either mode, finally, $\mathcal{A}$ outputs $b' \in \{0, 1\}$ for the guess of b. If $b = b'$ then $\mathcal{B}$ outputs 1 (meaning $Z = e(g, g)^{abs}$). Otherwise, it outputs 0 (meaning Z is random in $\mathbb{G}_T$).

We see that in both modes, the simulation is perfect. Furthermore, if $\boldsymbol{Y}$ is sampled from $\mathcal{R}_{BDH}$ then $\Pr[\mathcal{B}(\boldsymbol{Y}) = 0] = \frac{1}{2}$. On the other hand, if $\boldsymbol{Y}$ is sampled from $\mathcal{P}_{BDH}$ then we have $|\Pr[\mathcal{B}(\boldsymbol{Y}) = 0] - \frac{1}{2}| \geq \epsilon$. It follows that $\mathcal{B}$ has advantage at least ϵ in solving the DBDH problem in $\mathbb{G}$. This concludes the proof.

6 Efficiency

6.1 Comparison to Schemes Secure under the DBDH Assumption

In this section, we compare efficiency among available revocable ABE schemes in which security is based on the DBDH assumption. This is shown in Table 2.

Table Description. Denote by $|\mathsf{sk}|, |\mathsf{ct}|, |\mathsf{uk}|, |\mathsf{pk}|$ the size of user secret key, ciphertext overhead, update key, and public key, respectively. The columns Normal, Direct revoke, and Indirect revoke show the communication cost in the respective modes. (Normal means no user is revoked). Here r is the number of revoked user. n is the number of all users. ℓ is the size of rows in the LSSS matrix, which is equal to the number of attributes presented in the access structure. k is the size of attribute set. m is maximum size allowed for k. d is maximum size of $\mathsf{Cover}(R)$ for all $R \subseteq \mathcal{U}$. All values in the table show the amount of group elements in $\mathbb{G}$. Symbol '-' denotes unavailability of that mode.

DBDH-based Directly Revocable ABE. As stated in Section §1.2, there is no available directly revocable ABE which is proved secure under the DBDH assumption yet. We briefly show how to construct one here. We use the methodology in [3] for combining broadcast encryption (BE) and ABE. In their paper, they combine BE of [8] and [21] to ABE of [16], yielding two efficient directly revocable ABE systems, albeit they are based on much stronger assumptions (n-BDHE and q-MEBDH respectively). If we use this methodology to combine BE of [10], in which security is based on DBDH, and ABE of [16], it will yield a DBDH-based directly revocable ABE with parameters in the Table 2 (Variant of [3]).

Comparison. From Table 2, one can see that our HR-ABE scheme has the key size roughly the same as both one-mode revocable ABEs (and hence half size of the trivial HR-ABE from their combination). Indeed only $2\log(n)$ is increased from both one-mode schemes. The ciphertext size in direct mode is the same as the

Table 2. Comparison among available (revocable) ABE schemes based on the DBDH assumption

Scheme	$\|\mathsf{sk}\|$	Normal	Direct revoke mode	Indirect revoke mode		$\|\mathsf{pk}\|$
		$\|\mathsf{ct}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{uk}\|$	
[16]	2ℓ	$k+1$	-	-	-	$m+3$
[6]	$2\ell \log(n)+1$	$k+2$	-	$k+2$	$2(r\log(\frac{n}{r})+r)$	$m+3$
[3](variant)	$2\ell \log(n)$	$k+2$	$k+r\log(\frac{n}{r})+r+1$	-	-	$m+d+3$
Ours	$2(\ell+1)\log(n)$	$k+2$	$k+r\log(\frac{n}{r})+r+1$	$k+2$	$2(r\log(\frac{n}{r})+r)$	$m+d+3$

Table 3. Efficiency of available directly revocable ABE based on stronger assumptions

Scheme	$\|\mathsf{sk}\|$	Normal $\|\mathsf{ct}\|$	Direct revoke mode $\|\mathsf{ct}\|$	$\|\mathsf{pk}\|$	Assumption
[3](scheme 1)	2ℓ	$k+2$	$k+2$	$m+2n+2$	n-BDHE
[3](scheme 2)	$2\ell+2$	$k+2$	$k+2r+1$	$m+7$	q-MEBDH

variant of [3]. The ciphertext and update key sizes in indirect mode is the same as that of [6]. Hence, we can conclude that among revocable ABE systems that are based on the DBDH assumption, ours is the most flexible since it allows two revocation modes and has efficiency roughly the same as the one-mode schemes.

6.2 Comparisons to Schemes Secure under Stronger Assumptions

In this section, we compare our HR-ABE to the trivially combined schemes between Boldyreva et al. [6] and two directly revocable schemes of Attrapadung-Imai [3], which are based on *much stronger* assumptions, namely the n-BDHE [8] and the q-MEBDH [21] assumptions. Both directly revocable ABE schemes have the private key size about 2ℓ. See Table 3 for their efficiency. Thus, both combined schemes have the private key size about $2\ell \log(n) + 2\ell$. Therefore, our HR-ABE scheme is still more efficient when $\log(n) \leq \ell$. Note also that the first combined scheme ([6] and the first scheme of [3]) is not so efficient in that its public key is large as being linear in n.

7 Concluding Remarks

We presented a formalization and a construction of hybrid revocable attribute-based encryption. An HR-ABE system allows senders to select whether to use either direct or indirect revocation mode when encrypting a message. With direct mode, the sender specifies the list of revoked users directly into the encryption algorithm. With indirect mode, the sender specifies just the encrypt time (besides a usual attribute set), while receivers obtain a key update material at each time slot to update their keys from the authority, albeit in such a way that only non-revoked users are able to update (hence revocation is enforced indirectly). Our HR-ABE scheme requires each receiver to store only one key, which is nonetheless

can be used to decrypt ciphertexts in either mode. The key size in our hybrid scheme is roughly the same as that of the two currently best one-mode revocable schemes. We proved the security for the selective-target and semi-static query model under the DBDH assumption.

As for future direction regarding revocable ABE, it would be interesting to obtain more efficient schemes (*e.g.,* with constant-size keys and/or ciphertexts, update keys). Another direction could be to obtain revocable ABE schemes (one-mode or hybrid) which are secure in the adaptive security model.

References

1. Aiello, W., Lodha, S., Ostrovsky, R.: Fast digital identity revocation (extended abstract). In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 137–152. Springer, Heidelberg (1998)
2. Attrapadung, N., Imai, H.: Dual-policy attribute based encryption. In: Abdalla, M., Pointcheval, D., Fouque, P.A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 168–185. Springer, Heidelberg (2009)
3. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute-based encryption. In: Boyen, X., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 248–265. Springer, Heidelberg (2009)
4. Libert, B., Vergnaud, D.: Adaptive-ID secure revocable identity-based encryption. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 1–15. Springer, Heidelberg (2009)
5. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy 2007, pp. 321–334 (2007)
6. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: ACM Conference on Computer and Communications Security 2008, pp. 417–426 (2008)
7. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
8. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
9. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)
10. Dodis, Y., Fazio, N.: Public-key broadcast encryption for stateless receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)
11. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
12. Gentry, C.: Certificate-based encryption and the certificate revocation problem. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 272–293. Springer, Heidelberg (2003)
13. Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with short ciphertexts). In: Joux, A. (ed.) Eurocrypt 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009)
14. Gollé, P., Staddon, J., Gagne, M., Rasmussen, P.: A content-driven access control system. In: Symposium on Identity and Trust on the Internet — IDtrust 2008, pp. 26–35 (2008)

15. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute-based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
16. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security 2006, pp. 89–98 (2006)
17. Micali, S.: Efficient certificate revocation. Tech. Report MIT/LCS/TM-542b (1996)
18. 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)
19. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security 2007, pp. 195–203 (2007)
20. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
21. Sahai, A., Waters, B.: Revocation systems with very small private keys. Cryptology ePrint archive: report 2008/309
22. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. Cryptology ePrint archive: report 2008/290

A Some Further Discussions

A.1 On the Syntax Definition of HR-ABE

In this section, we provide some comparisons on the syntax definition between ours and that of Boldyreva et al. [6].

What to Be Revoked: Key vs Attributes. In a normal ABE system, the private key will be associated with an access structure $\mathbb{A}$. In general, the private key generation is a randomized algorithm, hence many concrete private keys will correspond to one $\mathbb{A}$. There are then two approaches for meaningfully defining revocation: (1) to revoke a specific one of many private keys that corresponds to $\mathbb{A}$ or (2) to revoke the access structure $\mathbb{A}$ itself (*i.e.,* to revoke all possible private keys that correspond to $\mathbb{A}$). Our definition in this paper is of the first type, while the one defined by Boldyreva et al. [6] seems to be of the second type.[2]

We feel that the type 1 definition is more useful since in practice, the revocation problem is motivated from the scenario where one specific key is leaked to some attacker. Indeed, we do not wish to revoke the access structure $\mathbb{A}$ itself from being able to use ever again in the system.

To meaningfully defining syntax of type 1, we must associate another identity dimension, which we call a serial number (or user index) denoted by id, uniquely to each possible private key for $\mathbb{A}$. When revoking, a revocator will specify id

[2] This is only if we guess correctly, since they did not explicitly define the syntax for revocable ABE case in their paper. They did only for revocable IBE case and in this case their formalization is of the second type: to revoke all possible private keys for an identity.

instead of $\mathbb{A}$ itself to distinguish this specific key among all possible keys for $\mathbb{A}$. More generally, a revocator will specify a set of serial numbers he wants to revoke; any of these may or may not correspond to the same access structures. We capture this type of revocation in our definition in Section §3.1.

On the other hand, Boldyreva et al. [6] provided a definition of the type 2 (for their indirectly revocable IBE but implicitly for the ABE case). In such a system, a revocator will specify the access structure $\mathbb{A}$ he wants to revoke. This kind of definition implies that after being revoked, any private keys for $\mathbb{A}$ (even with different randomness) cannot be used for decryption in the system anymore.

Although the notion of serial number is not included in the syntax definition of [6], it is introduced in their proposed construction and is utilized in a similar way as our explicit use of id in our syntax definition. Hence, the information on the serial number in their scheme is maintained *internally* inside the state of the key authority. Therefore, to be able to include such kind of schemes in the syntax definition, they define the algorithms as stateful type ones. We discuss this issue below.

Algorithms: Stateless vs Stateful. In our main syntax definition of HR-ABE (described in Section §3.1), all algorithms are stateless. We also mentioned a simple augment definition with the stateful $\mathsf{KeyGen}^{\mathsf{s}}$ algorithm in order to maintain the assignment of unique id to each key in the real usage.

In contrast, Boldyreva et al. [6] formalized their (indirectly revocable) system as a stateful one in a little bit more complex way. Indeed, their formalization takes into account the notion of time order. In particular, the revoked set R at the present time will be depended on the time t and on all previous possible actions (such as revocation done previously).

We feel that our explicit stateless formalization is more useful in two aspects. First, since the revoked set R is independent from time, an instant or temporary revocation is explicitly possible. Second, it does not take into account the notion of time order, hence t could be any token identity and the scheme will still work.

A.2 On the Security Definition of HR-ABE

We compare our security notion to that of [6] here (for the indirect mode). First, the query model is essentially the same but since in [6] the order of time is important, hence their notion has to check the condition for example $t \leq t^{\star}$ when some queries are asked. In ours, t is an independent attribute, we thus only care about at the target $t^{\star}$. Second, their notion separates the revocation query and update key query and links them via the internal state. In ours, the revoke set is stated explicitly in the update key query, hence ours is simpler. Third, they only provide an explicit notion for the IBE case but leave the detail for the ABE case to the reader. Finally, ours seems to capture a stronger notion in the sense that the serial number id can be chosen by the adversary when query the $\mathcal{SK}$ oracle. (The last one is due to the difference of the syntax definition types described in Appendix §A.1, though).