Outsider-Anonymous Broadcast Encryption with Sublinear Ciphertexts

Nelly Fazio[1,2] and Irippuge Milinda Perera[2]

[1] The City College of CUNY, fazio@cs.ccny.cuny.edu
[2] The Graduate Center of CUNY, {nfazio,iperera}@gc.cuny.edu

Abstract. In the standard setting of broadcast encryption, information about the receivers is transmitted as part of the ciphertext. In several broadcast scenarios, however, the identities of the users authorized to access the content are often as sensitive as the content itself. In this paper, we propose the first broadcast encryption scheme with sublinear ciphertexts to attain meaningful guarantees of receiver anonymity. We formalize the notion of *outsider-anonymous broadcast encryption*(oABE), and describe generic constructions in the standard model that achieve outsider-anonymity under adaptive corruptions in the chosen-plaintext and chosen-ciphertext settings. We also describe two constructions with enhanced decryption, one under the gap Diffie-Hellman assumption, in the random oracle model, and the other under the decisional Diffie-Hellman assumption, in the standard model.

Keywords: Recipient Privacy, Broadcast Encryption, Anonymous IBE, Subset Cover Framework.

1 Introduction

Conventional encryption provides the means for secret transmission of data in point-to-point communication. The setting of broadcast encryption [1, 2], instead, consists of a *sender*, an insecure unidirectional *broadcast channel*, and a universe of *receivers*. When the sender wants to transmit some digital content, it specifies the set of authorized receivers and creates an encrypted version of the content. A secure broadcast encryption scheme enables legitimate receivers to recover the original content, while ensuring that excluded users just obtain meaningless data, even in the face of collusions.

The intrinsic access control capabilities of broadcast encryption schemes make them a useful tool for many natural applications, spanning from protecting copyrighted content distributed as stored media [3], to managing digital subscriptions to satellite TV, to controlling access in encrypted file systems [4]. Thanks to its versatility, broadcast encryption has received a lot of attention from the crypto research community in recent years (see *e.g.*, [5–14]). The quest, however, has been for ever more efficient solutions in terms of broadcast communication, key storage and encryption/decryption running time. Little attention, instead, has been devoted to the exploration of refined security models that accurately account for the requirements inherent in multi-recipient communication. More specifically, the focus has been on providing assurance for sender-oriented properties, while overlooking the security and privacy concerns of the receivers.

One problem with the above (informal) definition of broadcast encryption is the implicit requirement that, whenever the digital content is encrypted and sent in broadcast, information about the set of authorized receivers is necessary to decrypt it correctly. Therefore, the set of authorized receivers is transmitted as part of the ciphertext. This in particular implies that an eavesdropper, even if unable to recover the message, can still easily discover the identities of the actual receivers of the content. A way to address the privacy implications that result from specifying explicitly the set of authorized receivers in the broadcast is to use ephemeral IDs and to keep secret the table that associates such IDs with the actual receivers. This simple solution, however, would at best result in a pseudonym system, in which it is still possible to link pseudonyms across transmissions and determine whether the same entity is an authorized receiver for two different broadcasts.

Anonymous Broadcast Encryption. An interesting variant of the broadcast encryption setting was proposed by Barth *et al.* in [15]. Therein, the authors introduce the notion of *private* broadcast encryption scheme, explicitly aiming to protect the identities of the receivers. As a proof-of-concept, they also suggest both generic and number-theoretic public-key constructions that do not leak any information about the list of authorized receivers, and are secure in the standard model and in the random oracle model, respectively. The proposed schemes, however, have communication complexity linear in the number of recipients. In [16], Libert *et al.* recently suggested proof techniques to argue the security of (a variant of) the number-theoretic construction of [15] without reliance on random oracles, thus attaining anonymous broadcast encryption with efficient decryption in the standard model. Still, ciphertexts in the resulting construction have length linear in the number of recipients.

Krzywiecki *et al.* presented a private public-key broadcast encryption scheme with communication complexity proportional to the number of revoked users [17]. The security analysis of the proposed solution is rather informal, however, so the security guarantees are at best heuristic.

In [18], Yu *et al.* presented the first *secret-key* multicast scheme with membership anonymity and communication complexity independent of the number of receivers. The proposed scheme not only hides the *identities* of the receives, but also *the number* of users allowed to receive the content. A shortcoming is that only a single user can be revoked for each broadcast.

A promising research line toward practical receiver-anonymous broadcast encryption has recently been started by Jarecki and Liu [19]. The authors propose the first construction of an efficient unlinkable secret handshake scheme, which is an authenticated key exchange protocol providing *affiliation/policy hiding* (*i.e.*, the transmission hides the affiliation and the identities of all parties) and *unlinkability* (*i.e.*, it is impossible to link any two instances of the secret handshake protocol). The proposed construction can be seen as a *stateful* version of a public-key broadcast encryption scheme, with the additional property of protecting the receivers' identities. Statefulness, however, implies that the key used to encrypt the broadcasts changes for each transmission, and receivers need to keep track of the changes to be able to recover the content.

An interesting trait of the of construction of [19] is that it trades some degree of anonymity for better efficiency: while the receiver's identities are hidden from outsiders, the scheme still allows authorized users to learn information about other members of the receiver set.

Our Contributions. In this paper we propose the first broadcast encryption scheme with sublinear ciphertexts to achieve meaningful guarantees of receiver anonymity. In particular, we formalize the notion of *outsider-anonymous broadcast encryption*(oABE), and describe a generic construction based on any anonymous identity-based encryption scheme (AIBE). Compared with the work of [19], our construction has the advantage of being *stateless*, and with constant public key size.

Additionally, by adapting the techniques of [15], we also obtain an efficient construction with enhanced decryption, where for a given oABE ciphertext, the decryption algorithm executes a single AIBE decryption operation. As outlined in Table 1, by relaxing the anonymity guarantees, our constructions achieve sublinear ciphertexts size and constant public key size.

Organization. Section 2 provides a brief review of the Subset Cover Framework [6] and of Anonymous Identity-Based Encryption [20, 21]. The setting of outsider-anonymous broadcast encryption is introduced in Sect. 3. In Sect. 4 we first present generic constructions in the standard model that achieve outsider-anonymity under adaptive corruptions in the chosen-plaintext (Sect. 4.1) and

Table 1. Comparison of the main efficiency parameters of our oABE scheme with [15] and [16]. Our construction trades full anonymity (achieved by [15, 16]) for sublinear ciphertexts and constant public key size.

	Scheme	PK Length	SK Length	CT Length	Decryption Attempts
Regular	BBW06 [15]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	$\mathcal{O}(N-r)$	$\mathcal{O}(N-r)$
	LPQ12 [16]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	$\mathcal{O}(N-r)$	$\mathcal{O}(N-r)$
	Ours (oABE)	$\mathcal{O}(1)$	$\mathcal{O}(\log N)$	$\mathcal{O}\left(r\log\left(\frac{N}{r}\right)\right)$	$\mathcal{O}\left(r\log\left(\frac{N}{r}\right)\log N\right)$
Enhanced	BBW06 [15]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	$\mathcal{O}(N-r)$	1
	LPQ12 [16]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	$\mathcal{O}(N-r)$	1
	Ours (oABE)	$\mathcal{O}(N)$	$\mathcal{O}(\log N)$	$\mathcal{O}\left(r\log\left(\frac{N}{r}\right)\right)$	1

chosen-ciphertext (Sect. 4.2) settings. Next, we describe a CCA-secure construction with enhanced decryption (Sect. 4.3) under the gap Diffie-Hellman assumption in the random oracle model, and outline how to extend it to the standard model, using the twin-DH-based techniques of [16]. Finally, we outline an optimization for the symmetric-key setting to accommodate storage-sensitive systems and attain constant key storage at the Center, while maintaining efficient decryption and logarithmic storage at the receivers (Sect. 4.4).

2 Background

2.1 The Subset Cover Framework

The *Subset Cover Framework* proposed by Naor et al. [6] is an environment for defining and analyzing the security of revocation schemes in the symmetric key setting, where only the Center can broadcast. The main idea of this framework is to define a collection $\mathcal{S}$ of subsets of the universe of users $\mathcal{U} = \{1, \ldots, N\}$ in the system, and assign each subset $S_j \in \mathcal{S}$ a long-lived key, which is also provided to the users belonging to S_j. When broadcasting a message m, first the Center determines the set of revoked users $\mathcal{R}$, then it finds a set of disjoint subsets $\mathcal{C}$ from the collection $\mathcal{S}$ that "covers" the set $\mathcal{U}\backslash\mathcal{R}$ of receivers, and finally it encrypts the short-lived session key used to encrypt m under all the long-lived keys associated with each subset in $\mathcal{C}$.

In [6], the authors also provide two instantiations of revocation schemes in the Subset Cover framework namely, the Complete Subtree (CS) method and the Subset Difference (SD) method. In the CS method, the key assignment is information-theoretic but the ciphertext is $\mathcal{O}\left(r\log\left(\frac{N}{r}\right)\right)$ long, whereas in the SD method, the ciphertext length is $\mathcal{O}(2r-1)$ but the key assignment is computational, where r is the number of revoked users. Although the ciphertext length of the CS method is asymptotically bigger than that of the SD method, we are still interested in the CS method due to its information-theoretic key assignment nature, which seems to be crucial for efficiently preserving the anonymity of the receivers.

Complete Subtree (CS) Method. In the Complete Subtree (CS) method as introduced in [6], the N users in the system are represented as the leaves of a full binary tree $\mathcal{T}$. Since this requires N to be a power of 2, dummy users are added to the system in case N is not a power of 2. The collection $\mathcal{S}$ contains all possible complete subtrees of $\mathcal{T}$. More precisely, $\mathcal{S}$ contains a subtree for every node $v_j \in \mathcal{T}$. Since there are $2N-1$ nodes in $\mathcal{T}$, $|\mathcal{S}| = 2N-1$.

As for key assignment, every subtree in $\mathcal{S}$ is assigned a long-lived symmetric key which is also made available to the users (leaves) of the given subtree. Since any user u_i, for $1 \leq i \leq N$, is a member of all the subtrees rooted at each node v_j, for $1 \leq j \leq \log N + 1$, in the path from the root of $\mathcal{T}$ down to u_i, the length of the user secret key is $\mathcal{O}(\log N)$.

The ciphertext length in the CS method is $\mathcal{O}\left(r \log\left(\frac{N}{r}\right)\right)$ due to the fact that a logarithmic number of subtrees is required to exclude each of the r revoked users (see [6] for further details).

Extension of the CS Method to the Public Key Setting. As mentioned earlier, the original CS method applies in the symmetric key setting. Thus, only the Center can broadcast since only it knows all the long-lived keys associated with each subtree in $\mathcal{S}$. In [8], Dodis and Fazio extended the original CS method to the public key setting by using a two step process.

The first step is a unique assignment of hierarchical identifiers (HID) to the nodes in $\mathcal{T}$ as follows. First, assign the root of $\mathcal{T}$ a special ID, which we refer to as `Root`. Then, assign each edge of $\mathcal{T}$ with ID 0 or 1 depending on whether the edge connects its parent node to the left or right child. Now, HID_j of any node $v_j \in \mathcal{T}$ can be computed by concatenating all the edge IDs starting from the root of $\mathcal{T}$ down to v_j and then pre-pending the root ID at the front. Since any prefix of HID_j of v_j represents the valid HID of a parent node of v_j, for the simplicity of notation, we denote by $\mathsf{HID}_{i|j}$ the prefix of the hierarchical identifier HID_i of length j.

The second step is to use Identity-Based Encryption (IBE), further explained in Sect. 2.2, to encrypt the short-lived session key during broadcast, essentially porting the original CS method to the public key setting. This allows any user to broadcast a message since the tree structure of the users $\mathcal{T}$ and the HIDs of the roots of the subtrees of $\mathcal{T}$ are publicly known. In this setting, the Center acts as the trusted authority to provide each user with the $\log N + 1$ IBE secret keys of the HIDs of the roots of the subtrees that the user belongs to.

2.2 Anonymous Identity-Based Encryption (AIBE)

Identity-Based Encryption (IBE), originally proposed by Shamir in [22], is a public key encryption scheme in which the user public key is an arbitrary bit-string and the user secret key is generated by a trusted authority known as the *Private Key Generator* (PKG) using its master key. The first implementation of this scheme was given in [23] (further implementations can be found in [24–26] to name a few).

An IBE scheme is called anonymous, formally called Anonymous Identity-Based Encryption (AIBE), if an adversary cannot distinguish the public key under which a ciphertext is generated. This notion of anonymity was first introduced in [20]. Subsequent implementations can be found in [27] and [21]. Given below is the formal definition of an AIBE scheme. We refer the reader to [20] for further details including the formal definition of security.

Definition 1. *An anonymous identity-based encryption* (AIBE) *scheme, associated with a message space $\mathcal{MSP}$, and a ciphertext space $\mathcal{CSP}$, is a tuple of probabilistic polynomial algorithms* $(\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

$(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathsf{Init}(1^\lambda)$**:** *The initialization algorithm* Init *takes as input the security parameter 1^λ, and outputs the public key PK and the master secret key MSK of the system.*

$sk_{\mathsf{ID}} \leftarrow \mathsf{Ext}(\mathrm{PK}, \mathrm{MSK}, \mathsf{ID})$**:** *The key extraction algorithm* Ext *takes as input the public key PK, the master secret key MSK, and an identifier $\mathsf{ID} \in \{0,1\}^*$. It outputs the secret key sk_{ID} capable of decrypting ciphertexts intended for the holder of the given identifier* ID.

$c \leftarrow \mathsf{Enc}(\mathrm{PK}, \mathsf{ID}, m)$**:** *The encryption algorithm* Enc *algorithm takes as input the public key PK, an identifier $\mathsf{ID} \in \{0,1\}^*$, and a message $m \in \mathcal{MSP}$. It then outputs a ciphertext $c \in \mathcal{CSP}$.*

$m/\bot := \mathsf{Dec}(\mathrm{PK}, sk_{\mathsf{ID}}, c)$: *Given the public key PK, a secret key* sk_{ID}*, and a ciphertext* $c \in \mathcal{CSP}$*, the decryption algorithm* Dec *either outputs a message* $m \in \mathcal{MSP}$ *or the failure symbol* $\bot$*. We assume that* Dec *is deterministic.*

Correctness. For every $\mathsf{ID} \in \{0,1\}^*$ and every $m \in \mathcal{MSP}$, if sk_{ID} is the secret key output by $\mathsf{Ext}(\mathrm{PK}, \mathrm{MSK}, \mathsf{ID})$, then $\mathsf{Dec}(\mathrm{PK}, sk_{\mathsf{ID}}, \mathsf{Enc}(\mathrm{PK}, \mathsf{ID}, m)) = m$.

Weakly Robust AIBE. The *Robust Encryption*, formalized by Abdalla et al. [28], requires that it is hard to produce a ciphertext that is valid for two different users. In [28], the authors define two types of robustness, strong and weak. Informally, an AIBE scheme is called *weakly robust*, if any adversary has negligible advantage in producing two identities $\mathsf{ID}_0, \mathsf{ID}_1$ and a message m such that the encryption of m under ID_0 can be decrypted with the private key associated with ID_1 leading to a non-$\bot$ result. In [28], the authors also provide a transformation algorithm which makes possible to obtain a weakly robust AIBE scheme from a regular AIBE one.

3 Outsider-Anonymous Broadcast Encryption (oABE)

3.1 The Setting

Definition 2. *An outsider-anonymous broadcast encryption* (oABE) *scheme, associated with a universe of users* $U = \{1, \ldots, N\}$*, a message space* $\mathcal{MSP}$*, and a ciphertext space* $\mathcal{CSP}$*, is a tuple of probabilistic polynomial algorithms* (Setup, KeyGen, Encrypt, Decrypt) *such that:*

$(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, N)$: *The* Setup *algorithm takes as input the security parameter* 1^λ *and the number of users in the system* N*. It outputs the public key PK and the master secret key MSK of the system.*

$sk_i \leftarrow \mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$: *The key generation algorithm* KeyGen *takes as input the public key PK, the master secret key MSK, and a user* $i \in U$*. It outputs the secret key* sk_i *of the user* i*.*

$c \leftarrow \mathsf{Encrypt}(\mathrm{PK}, S, m)$: *The* Encrypt *algorithm takes as input the public key PK, the set of receivers* $S \subseteq U$*, and a message* $m \in \mathcal{MSP}$*. It then outputs a ciphertext* $c \in \mathcal{CSP}$*.*

$m/\bot := \mathsf{Decrypt}(\mathrm{PK}, sk_i, c)$: *Given the public key PK, a secret key* sk_i*, and a ciphertext* $c \in \mathcal{CSP}$*, the* Decrypt *algorithm either outputs a message* $m \in \mathcal{MSP}$ *or the failure symbol* $\bot$*. We assume that* Decrypt *is deterministic.*

Correctness. For every $S \subseteq U$, every $i \in S$, and every $m \in \mathcal{MSP}$, if sk_i is the secret key output by $\mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$ then $\mathsf{Decrypt}(\mathrm{PK},\ sk_i,\ \mathsf{Encrypt}(\mathrm{PK},\ S,\ m)) = m$.

Notice that the decryption algorithm in the above definition does not require the set of recipients S as an input. We stress that this is crucial for providing any level of anonymity in a broadcast encryption scheme.

3.2 The Security Model

Degrees of Anonymity. The degree of recipient-set anonymity captured in our security model, which we call *outsider-anonymity*, lies between the complete lack of protection that characterizes traditional broadcast encryption schemes as introduced in [2,14], and the full anonymity provided in schemes such as [15,16]. In an oABE scheme, when the adversary receives a ciphertext of which she is not a legal recipient, she will be unable to learn anything about the identities of the legal recipients (let alone the contents of the ciphertext). Still, for those ciphertexts for which the adversary is in the authorized set of recipients, she might also learn the identities of some the other legal recipients. This seems a natural relaxation, since often the *contents* of the communication already reveals something about the recipient set. At the same time, our new intermediate definition of security might allow the construction of more efficient anonymous broadcast encryption schemes; for example, in Sect. 4 we describe the first broadcast encryption scheme with sub-linear ciphertexts that attains some meaningful recipient-set anonymity guarantees.

CCA Security. We now present the security requirements for a broadcast encryption scheme to be *outsider-anonymous*against chosen-ciphertext attacks (CCA). First we define the CCA of an oABE scheme as a game, which we term oABE-IND-CCA, played between a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The security requirement is that $\mathcal{A}$'s advantage of winning the oABE-IND-CCA game is negligible. The high-level idea of this game is for any two sets of recipients $S_0, S_1 \in U$, $\mathcal{A}$ cannot distinguish between a ciphertext intended for the recipient set S_0 and a ciphertext intended for the recipient set S_1 given the fact that the $\mathcal{A}$ does not possess the secret key of any user in $S_0 \cup S_1$. We require the two sets S_0, S_1 be the same size in order to avoid trivial attacks. The formal definitions follow.

Definition 3. *The* oABE-IND-CCA *game defined for an* oABE *scheme* $\mathit{\Pi} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$*, a* PPT *adversary* $\mathcal{A}$*, and a challenger* $\mathcal{C}$ *is as follows:*

Setup: *$\mathcal{C}$ runs $(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, N)$ and gives $\mathcal{A}$ the resulting public key PK, keeping the master secret key MSK to itself. $\mathcal{C}$ also initializes the set of revoked users* Rev *to be empty.*

Phase 1: *$\mathcal{A}$ adaptively issues queries $q_1, \ldots, q_m$ where each q_i is one of the following:*

- *Secret key query i: $\mathcal{A}$ requests the secret key of the user $i \in U$.*
  $\mathcal{C}$ runs $sk_i \leftarrow \mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$ to generate the secret key sk_i of the user i, adds i to Rev*, and sends sk_i to $\mathcal{A}$.*
- *Decryption query (i, c): $\mathcal{A}$ issues a decryption query where $i \in U$ and $c \in \mathcal{CSP}$. First, $\mathcal{C}$ runs $sk_i \leftarrow \mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$ to generate the secret key sk_i of the user i. Then, it runs $\mathsf{Decrypt}(\mathrm{PK}, sk_i, c)$ and gives the output to $\mathcal{A}$.*

Challenge: *$\mathcal{A}$ gives $\mathcal{C}$ two equal length messages $m_0, m_1 \in \mathcal{MSP}$ and two equal length sets of user identities $S_0, S_1 \subseteq U$ with the restriction that $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$. $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$, runs $c^* \leftarrow \mathsf{Encrypt}(\mathrm{PK}, S_b, m_b)$, and sends c^* to $\mathcal{A}$.*

Phase 2: *$\mathcal{A}$ adaptively issues additional queries $q_{m+1}, \ldots, q_n$ where each q_i is one of the following:*

- *Secret key query i such that $i \notin S_0 \cup S_1$.*
- *Decryption query (i, c) such that, if $i \in S_0 \cup S_1$, then $c \neq c^*$.*

In both cases, $\mathcal{C}$ responds as in Phase 1.

Guess: *$\mathcal{A}$ output a guess $b' \in \{0, 1\}$ and wins if $b' = b$.*

We refer to such an adversary $\mathcal{A}$ as an oABE-IND-CCA *adversary. The advantage of $\mathcal{A}$ winning the above game is defined as,*

$$\mathsf{Adv}_{\mathcal{A},\mathit{\Pi}}^{\mathsf{oABE\text{-}IND\text{-}CCA}} = \left|\Pr\left[b' = b\right] - \tfrac{1}{2}\right|$$

The probability is over the random bits used by the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$.

Definition 4. *An* oABE *scheme* $\mathit{\Pi} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is* $(t,\ q_{sk},\ q_d,\ \epsilon)$*-secure if for any t-time* oABE-IND-CCA *adversary $\mathcal{A}$ making at most q_{sk} chosen secret key queries and at most q_d chosen decryption queries, we have that $\mathsf{Adv}_{\mathcal{A},\mathit{\Pi}}^{\mathsf{oABE\text{-}IND\text{-}CCA}} \leq \epsilon$. As a shorthand, we say that $\mathit{\Pi}$ is $(t, q_{sk}, q_d, \epsilon)$-*oABE-IND-CCA *secure.*

CPA Security. The chosen plaintext attack (CPA) of an oABE scheme is defined similar to the $\mathsf{oABE\text{-}IND\text{-}CCA}$ game with the restriction that the adversary is not allowed to issue any decryption queries during *Phase 1* and *Phase 2*. The adversary is still allowed to issue secret key queries. The CPA security game is termed $\mathsf{oABE\text{-}IND\text{-}CPA}$.

Definition 5. *An* oABE *scheme* $\mathit{\Pi} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is* $(t,\, q_{sk},\epsilon)$-$\mathsf{oABE\text{-}IND\text{-}CPA}$ *secure if* $\mathit{\Pi}$ *is* $(t, q_{sk}, 0, \epsilon)$-$\mathsf{oABE\text{-}IND\text{-}CCA}$ *secure.*

Remark 1. Our definition of security of an outsider-anonymous broadcast encryption scheme can be easily transformed to a definition of security of a fully anonymous broadcast encryption scheme by changing the restriction in the challenge phase, which is currently $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$, to $\mathsf{Rev} \cap (S_0 \triangle S_1) = \emptyset$.[3]

4 Our Constructions

We now present a CPA secure construction and two CCA secure constructions of outsider-anonymous broadcast encryption (oABE) schemes. In a nutshell, the key point of our constructions is to combine an anonymized version of the public-key extension by Dodis and Fazio [8] of the CS method by Naor et al. [6] with a fully secure weakly robust AIBE scheme such as [21]. Notice that our approach can be seen as a *framework* for achieving an oABE scheme by using any weakly robust AIBE scheme as an underlying primitive.

The ciphertext length in all constructions is $\mathcal{O}\left(r \log\left(\frac{N}{r}\right)\right)$ times the ciphertext length of the underlying AIBE scheme, and the user secret key length is $\mathcal{O}\left(\log N\right)$ times the user secret key length of the underlying AIBE scheme, where r is the number of revoked users and N is the total number of users in the system.

We provide two generic public-key constructions: a CPA secure construction in Sect. 4.1 and a CCA secure construction in Sect. 4.2. The limitation with both of these constructions is that on average, the $\mathsf{Decrypt}$ algorithm attempts $\mathcal{O}\left(r \log\left(\frac{N}{r}\right) \log N\right)$ decryption operations of the underlying AIBE scheme. In Sect. 4.3, we present an optimized CCA secure construction in which for a given oABE ciphertext, the $\mathsf{Decrypt}$ algorithm executes a single AIBE decryption operation.

4.1 A Generic CPA Public-Key Construction

Given a weakly robust AIBE scheme $\mathit{\Pi}' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$, we construct an oABE scheme $\mathit{\Pi} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as follows. Let $\mathcal{T}$ denote the binary tree of N users in the system with respect to the CS method. For simplicity, we assume below that $N = 2^n$.

$\mathsf{Setup}(1^\lambda, N)$**:** Obtain $(\mathrm{PK}', \mathrm{MSK}') \leftarrow \mathsf{Init}(1^\lambda)$. Output the PK and MSK as follows,

$$\mathrm{PK} = (\mathrm{PK}', N) \qquad \mathrm{MSK} = \mathrm{MSK}'$$

$\mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$**:** Let $\mathsf{HID}_i = (\mathtt{Root}, \mathsf{ID}_1, \ldots, \mathsf{ID}_n)$ be the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. For $j = 1$ to $n + 1$, compute $sk_{i,j} \leftarrow \mathsf{Ext}(\mathrm{PK}', \mathrm{MSK}', \mathsf{HID}_{i|j})$. Output the secret key sk_i of the user i as follows,

$$sk_i = (sk_{i,1}, \ldots, sk_{i,n+1})$$

[3] For any two sets S_0, S_1, their symmetric difference is denoted by $S_0 \triangle S_1$.

$\mathsf{Encrypt}(\mathrm{PK}, S, m)$**:** Let Cover be the family of subtrees covering the set of receivers S according to the CS method. For each subtree T_j in Cover, let HID_j be the hierarchical identifier associated with the root of T_j. Let $l = |\mathsf{Cover}|$, $r = N - |S|$ and $L = \lfloor r \log\left(\frac{N}{r}\right) \rfloor$.

For $1 \leq j \leq l$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j, m)$. Choose $\widetilde{m} \overset{\$}{\leftarrow} \mathcal{MSP}$.

For $l + 1 \leq j \leq L$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$, where dummy is a special identifier used to obtain padding ciphertext components. Output the ciphertext c as follows,

$$c = \left(c_{\pi(1)}, \ldots, c_{\pi(L)}\right)$$

where $\pi : \{1, \ldots, L\} \rightarrow \{1, \ldots, L\}$ is a random permutation.[4]

$\mathsf{Decrypt}(\mathrm{PK}, sk_i, c)$**:** Parse the secret key sk_i as the tuple $(sk_{i,1}, \ldots, sk_{i,n+1})$ and the ciphertext c as the tuple $(c_1, \ldots, c_L)$.

For $k = 1$ to $n + 1$,

1. For $j = 1$ to L,
   (a) Compute $m \leftarrow \mathsf{Dec}(\mathrm{PK}', sk_{i,k}, c_j)$.
   (b) If $m \neq \perp$, return m. Otherwise, continue to next j.
2. If $k = n + 1$, return $\perp$. Otherwise, continue to next k.

The correctness of this CPA secure generic public-key construction follows from the correctness of the underlying AIBE scheme. In Theorem 1 (whose proof is provided in Appendix A.1), we establish the security of the above generic public-key construction based on the security of the underlying AIBE scheme.

Theorem 1. *If $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ is (t, q_{sk}, ϵ)-AIBE-IND-CPA secure, then the above construction is $\left(t, q_{sk}, 2\,\epsilon\, r \log\left(\frac{N}{r}\right)\right)$-oABE-IND-CPA secure.*

Parameters. When the above construction is instantiated with Gentry's Fully Secure IBE scheme in the CPA setting [21], we obtain the following parameter lengths. MSK is just one element in $\mathbb{Z}_p$ and the integer N. PK is only 3 group elements in $\mathbb{G}$. The user secret key consists of $(\log N + 1)$ elements in $\mathbb{Z}_p$ and $(\log N + 1)$ elements in $\mathbb{G}$. The ciphertext consists of $\lfloor r \log\left(\frac{N}{r}\right) \rfloor$ elements in $\mathbb{G}$ and $2 \lfloor r \log\left(\frac{N}{r}\right) \rfloor$ elements in $\mathbb{G}_{\mathrm{T}}$. Also notice that the Enc algorithm in Gentry's AIBE scheme does not require any pairing computations since they can be pre-computed.

4.2 A Generic CCA Public-Key Construction

Given a weakly robust AIBE scheme $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ and a strongly existentially unforgeable one-time signature scheme $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, we construct an oABE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as follows. Let $\mathcal{T}$ denote the binary tree of N users in the system with respect to the CS method. For simplicity, we assume below that $N = 2^n$.

$\mathsf{Setup}(1^\lambda, N)$**:** Obtain $(\mathrm{PK}', \mathrm{MSK}') \leftarrow \mathsf{Init}(1^\lambda)$. Output the PK and MSK as follows,

$$\mathrm{PK} = (\mathrm{PK}', N) \qquad \mathrm{MSK} = \mathrm{MSK}'$$

[4] For the simplicity of exposition, our construction encrypts the actual message m. The ciphertext length could be further reduced by using a hybrid encryption where m is encrypted using a symmetric key encryption algorithm with a symmetric key k, and k is then encrypted using the oABE scheme.

$\mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$: Let $\mathsf{HID}_i = (\mathtt{Root}, \mathsf{ID}_1, \ldots, \mathsf{ID}_n)$ be the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. For $j = 1$ to $n + 1$, compute $sk_{i,j} \leftarrow \mathsf{Ext}(\mathrm{PK}', \mathrm{MSK}', \mathsf{HID}_{i|j})$. Output the secret key sk_i of the user i as follows,

$$sk_i = (sk_{i,1}, \ldots, sk_{i,n+1})$$

$\mathsf{Encrypt}(\mathrm{PK}, S, m)$: Generate $(\mathrm{VK}, \mathrm{SK}) \leftarrow \mathsf{Sig\text{-}Gen}(1^\lambda)$. Let Cover be the family of subtrees covering the set of receivers S according to the CS method. For each subtree T_j in Cover, let HID_j be the hierarchical identifier associated with the root of T_j.
Let $l = |\mathsf{Cover}|$, $r = N - |S|$ and $L = \left\lfloor r \log\left(\frac{N}{r}\right) \right\rfloor$.
For $1 \leq j \leq l$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j, \mathrm{VK}||m)$. Let $\widetilde{m}$ be a random string of the same length as $\mathrm{VK}||m$. For $l+1 \leq j \leq L$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$, where $\mathtt{dummy}$ is a special identifier used to obtain padding ciphertext components. Compute the ciphertext c as follows,

$$c = \left(c_{\pi(1)}, \ldots, c_{\pi(L)}\right)$$

where $\pi : \{1, \ldots, L\} \rightarrow \{1, \ldots, L\}$ is a random permutation.
Generate $\sigma \leftarrow \mathsf{Sign}(\mathrm{SK}, \mathrm{VK}||c)$, and output $C = \sigma||c$.

$\mathsf{Decrypt}(\mathrm{PK}, sk_i, C)$: Parse the secret key sk_i as the tuple $(sk_{i,1}, \ldots, sk_{i,n+1})$ and the ciphertext C as the tuple $\sigma||\,(c_1, \ldots, c_L)$.
For $k = 1$ to $n + 1$,

```
1. For $j = 1$ to $L$,
   (a) Compute $m' \leftarrow \mathsf{Dec}(\mathrm{PK}', sk_{i,k}, c_j)$.
   (b) If $m' \neq \perp$, parse $m' = \mathrm{VK}||m$, and return $m$ if $\mathsf{Vrfy}(\mathrm{VK}, \sigma, \mathrm{VK}||c)$. Otherwise, continue to next $j$.
2. If $k = n + 1$, return $\perp$. Otherwise, continue to next $k$.
```

The correctness of this CCA secure generic public-key construction follows from the correctness of the underlying Σ and AIBE schemes. Next, in Theorem 2 (whose proof is provided in Appendix A.2), we establish the security of the above generic public-key construction based on the security of the underlying Σ and AIBE schemes.

Theorem 2. *If* $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is* (t, ϵ_1)*-strongly existentially unforgeable and* $\Pi' = (\mathsf{Init},\ \mathsf{Ext},\ \mathsf{Enc},\ \mathsf{Dec})$ *is* $(t, q_{sk}, q_d, \epsilon_2)$-$\mathsf{AIBE\text{-}IND\text{-}CCA}$ *secure, then the above construction is* $(t,\ q_{sk},\ q_d,\ 2(\epsilon_1 + \epsilon_2)\, r \log\left(\frac{N}{r}\right))$-$\mathsf{oABE\text{-}IND\text{-}CCA}$ *secure.*

Parameters. The parameter lengths of the above construction when instantiated with Gentry's Fully Secure IBE scheme in the CCA setting [21] are as follows. MSK is one element in $\mathbb{Z}_p$ and the integer N. PK consists of 5 group elements in $\mathbb{G}$ and the definition of a hash function H from a family of universal one-way hash functions. The user secret key consists of $3(\log N + 1)$ elements in $\mathbb{Z}_p$ and $3(\log N + 1)$ elements in $\mathbb{G}$. The ciphertext consists of $\left\lfloor r \log\left(\frac{N}{r}\right) \right\rfloor$ elements in $\mathbb{G}$ and $3 \left\lfloor r \log\left(\frac{N}{r}\right) \right\rfloor$ elements in $\mathbb{G}_\mathrm{T}$. Similar to Gentry's CPA secure AIBE construction, the Enc algorithm in the CCA secure construction does not require any pairing computations since they can be pre-computed.

4.3 An Enhanced CCA Public-Key Construction

The main limitation of our generic public-key constructions is the running time of the decryption algorithm. As described in the opening paragraphs of Sect. 4, decryption amounts to performing $\mathcal{O}\left(r \log\left(\frac{N}{r}\right) \log N\right)$ AIBE decryption attempts on average. The root cause behind this limitation is the decryption process's inability to identify the correct AIBE ciphertext component efficiently. In this section, we describe an enhancement of our generic public-key construction under the gap Diffie-Hellman assumption, in the random oracle model. The main idea of this enhancement is to adapt the techniques of [15] to the structure of our ciphertexts and attach a unique tag to each AIBE ciphertext component of a given oABE ciphertext. With this optimization, the $\mathsf{Decrypt}$ algorithm is able to identify the correct AIBE ciphertext component via a linear search through the whole oABE ciphertext components, at which point a single AIBE decryption operation suffices to recover the original plaintext. This yields an asymptotic decryption time of $\mathcal{O}\left(r \log\left(\frac{N}{r}\right) \log N\right)$, but in fact this is in a sense an overestimate, since the cost of searching for the correct ciphertext component is much less than carrying out multiple decryption attempts.

Given a weakly robust AIBE scheme $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ and a strongly existentially unforgeable one-time signature scheme $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, we construct an optimized oABE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as follows. Let $\mathcal{T}$ denote the binary tree of N users in the system with respect to the CS method. For simplicity, we assume below that $N = 2^n$. Let $\mathbb{G} = \langle g \rangle$ be a group with prime order $q > 2^\lambda$ in which CDH is hard and DDH is easy, where g is a group generator. Let $H' : \mathbb{G} \to \{0,1\}^\lambda$ be a cryptographic hash function that will be modeled as a random oracle in the security analysis.

$\mathsf{Setup}(1^\lambda, N)$**:** Obtain $(\mathrm{PK}', \mathrm{MSK}') \leftarrow \mathsf{Setup}'(1^\lambda)$. For each node (with the hierarchical identifier HID) in $\mathcal{T}$, draw $a_{\mathsf{HID}} \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, and compute $y_{\mathsf{HID}} = g^{a_{\mathsf{HID}}}$. Output the PK and MSK as follows,

$$\mathrm{PK} = \left(\mathrm{PK}', N, \mathbb{G}, g, \{y_{\mathsf{HID}}\}_{\mathsf{HID} \in \mathcal{T}}\right) \qquad \mathrm{MSK} = \left(\mathrm{MSK}', \{a_{\mathsf{HID}}\}_{\mathsf{HID} \in \mathcal{T}}\right)$$

$\mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$**:** Let $\mathsf{HID}_i = (\mathtt{Root}, \mathsf{ID}_1, \ldots, \mathsf{ID}_n)$ be the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. For $j = 1$ to $n+1$, set $\overline{sk}_{i,j} = a_{\mathsf{HID}_{i|j}}$, and compute $sk_{i,j} \leftarrow \mathsf{Init}(\mathrm{PK}', \mathrm{MSK}', \mathsf{HID}_{i|j})$. Output the secret key sk_i of the user i as follows,

$$sk_i = \left(\left(\overline{sk}_{i,1}, sk_{i,1}\right), \ldots, \left(\overline{sk}_{i,n+1}, sk_{i,n+1}\right)\right)$$

$\mathsf{Encrypt}(\mathrm{PK}, S, m)$**:** Generate $(\mathrm{VK}, \mathrm{SK}) \leftarrow \mathsf{Sig\text{-}Gen}(1^\lambda)$. Let Cover be the family of subtrees covering the set of receivers S according to the CS method. For each subtree T_j in Cover, let HID_j be the hierarchical identifier associated with the root of T_j.

Let $l = |\mathsf{Cover}|$, $r = N - |S|$ and $L = \left\lfloor r \log\left(\frac{N}{r}\right)\right\rfloor$. Draw $s \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, and compute $\bar{c}_0 = g^s$.
For $1 \leq j \leq l$, compute $\bar{c}_j = H'(y^s_{\mathsf{HID}_j})$, $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j, \mathrm{VK}||y^s_{\mathsf{HID}_j}||m)$.

Let $\widetilde{m}$ be a random string of the same length as $\mathrm{VK}||\bar{c}_0||m$. For $l+1 \leq j \leq L$, set $s_j \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, and compute $\bar{c}_j = H'(g^{s_j})$, $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$, where $\mathtt{dummy}$ is a special identifier used to obtain padding ciphertext components. Compute the ciphertext c as follows,

$$c = \left(\bar{c}_0, \left(\bar{c}_{\pi(1)}, c_{\pi(1)}\right), \ldots, \left(\bar{c}_{\pi(L)}, c_{\pi(L)}\right)\right)$$

where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation. Generate $\sigma \leftarrow \mathsf{Sign}(\mathrm{SK}, \mathrm{VK}||c)$, and output $C = \sigma||c$.

$\mathsf{Decrypt}(\mathrm{PK}, sk_i, c)$**:** Parse the secret key sk_i as the tuple $((\overline{sk}_{i,1},\, sk_{i,1}),\, \ldots,\, (\overline{sk}_{i,n+1},\, sk_{i,n+1}))$ and the ciphertext C as the tuple $(\sigma||\, \overline{c}_0,\, (\overline{c}_1,\, c_1),\, \ldots,\, (\overline{c}_L,\, c_L))$.

```
1. For $k = 1$ to $n + 1$,
   (a) Compute $y_k = H'(\overline{c}_0^{\,\overline{sk}_{i,k}})$
2. Check whether $\exists\ k \in [1, n + 1]$, $\exists\ j \in [1, L]$ such that $y_k = \overline{c}_j$
   (a) If suitable $k, j$ exist, compute $m' \leftarrow \mathsf{Dec}(\mathrm{PK}', sk_{i,k}, c_j)$. Parse $m'$ as $\mathrm{VK}||x||m$ and return $m$ if $x = \overline{c}_0^{\,\overline{sk}_{i,k}}$ and $\mathsf{Vrfy}(\mathrm{VK}, \sigma, \mathrm{VK}||c)$.
   (b) Otherwise, return $\perp$.
```

Notice that the check in Step 2. can be performed in expected time $\mathcal{O}(n + L) = \mathcal{O}(L)$, *e.g.*, using a hash table $\mathrm{H_T}$ to compute the intersection between $\{y_k\}_{k\in[1,n+1]}$ and $\{\overline{c}_j\}_{j\in[1,L]}$ as follows:

```
a. Initialize $\mathrm{H_T}$ to be empty.
b. For $k = 1$ to $n + 1$
   – Insert $(y_k, k)$ in $\mathrm{H_T}$.
c. For $j \in 1$ to $L$
   – Look up an entry of the form $(\overline{c}_j, k)$ in $\mathrm{H_T}$. If found, return $k$.
```

Theorem 3. *If* $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is* (t, ϵ_1)*-strongly existentially unforgeable,* $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ *is* $(t, q_{sk}, q_d, \epsilon_2)$-AIBE-IND-CCA *secure, and CDH is* (t, ϵ_3)*-hard in* $\mathbb{G}$ *and DDH is efficiently computable in* $\mathbb{G}$*, then the above construction is* $(t,\, q_{sk},\, q_d,\, 2(\epsilon_1 + \epsilon_2 + \epsilon_3)\; r \log\left(\frac{N}{r}\right))$-oABE-IND-CCA *secure, in the random oracle model.*

Proof. The proof follows the same structure of the proof for Theorem 2, where the analogous arguments to Lemma 3 and Lemma 4 are augmented with techniques from [15]. We defer the details to the full version of the paper.

Remark 2. Using the twin Diffie-Hellman methodology [29] via techniques similar to [16], it is possible to modify the enhanced CCA construction of Sect. 4.3 to get an outsider-anonymous broadcast encryption scheme that is adaptively CCA secure, in the standard model, under the decisional Diffie-Hellman assumption. We defer the details to the full version of the paper.

4.4 An Enhanced CCA Symmetric-Key Construction

The enhanced CCA public key construction achieves a major performance gain in the Decrypt algorithm compared to the generic CCA construction, but it also changes the length of the public key from $\mathcal{O}(1)$ to $\mathcal{O}(N)$. This increase in public key length may not be a concern for many practical constructions, since the public key can be stored as a static data file on a server on the Internet and also in users' computers. Still, for the symmetric-key setting it is possible to accommodate storage-sensitive systems and attain constant key storage at the Center, while maintaining efficient decryption and logarithmic storage at the receivers.

In particular, recall from Sect. 2.1 that in the symmetric-key setting, only the Center can broadcast messages to the receivers. Thus, the $\mathcal{O}(N)$ information from which the tags for efficient decryption are created does not need to be published. Therefore, this information can be compressed into $\mathcal{O}(1)$ key storage using a standard trick based on any length-tripling pseudo-random number generator G (*cf. e.g.*, the SD method of Naor *et al.* [6]). In other words, the random exponents associated with the subtrees of $\mathcal{T}$ (*cf.* Sect. 4.3) are now pseudo-randomly generated from a single seed, by repeated invocations of G on the left or right third of the result of the previous iteration, based on the path to the root of the subtree at hand. Finally, upon reaching the subtree root, the middle third of the pseudorandom output is used to generate the required exponent.

5 Conclusions and Future Work

In this work, we introduced the notion of outsider-anonymity in the broadcast encryption setting and showed that it enables efficient constructions of broadcast encryption schemes with sublinear communication complexity and meaningful anonymity guarantees. It remains an interesting open problem to construct receiver-anonymous broadcast encryption schemes that at once afford full anonymity to the receivers and attain performance levels comparable to those of standard broadcast encryption systems.

Acknowledgments. Nelly Fazio's research was sponsored in part by NSF award #1117675 and by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defe nce or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. This project was also partially supported by PSC-CUNY Awards 63356-00 41 and 64578-00 42, jointly funded by The Professional Staff Congress and The City University of New York.

References

1. Berkovits, S.: How to broadcast a secret. In: Advances in Cryptology—EUROCRYPT '91. (1991) 535–541
2. Fiat, A., Naor, M.: Broadcast encryption. In: Advances in Cryptology—CRYPTO '93. (1993) 480–491
3. AACS: Advanced access content system. http://www.aacsla.com/
4. Goh, E.J., Shacham, H., Modadugu, N., Boneh, D.: SiRiUS: Securing remote untrusted storage. In: ISOC Network and Distributed Systems Security Symposium—NDSS '03. (2003) 131–145
5. Garay, J.A., Staddon, J., Wool, A.: Long-lived broadcast encryption. In: Advances in Cryptology—CRYPTO '00. (2000) 333–352
6. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Advances in Cryptology—CRYPTO '01. (2001) 41–62
7. Halevy, D., Shamir, A.: The LSD broadcast encryption scheme. In: Advances in Cryptology—CRYPTO '02. (2002) 47–60
8. Dodis, Y., Fazio, N.: Public-key broadcast encryption for stateless receivers. In: ACM Digital Rights Management—DRM '02. (2002) 61–80
9. Dodis, Y., Fazio, N.: Public-key trace and revoke scheme secure against adaptive chosen ciphertext attack. In: IACR Public Key Cryptography—PKC '03. (2003) 100–115
10. Dodis, Y., Fazio, N., Kiayias, A., Yung, M.: Scalable public-key tracing and revoking. In: Principles of Distributed Computing—PODC '03. (2003) 190–199 Invited to the PODC '03 Special Issue of Journal of Distributed Computing.
11. Yao, D., Fazio, N., Dodis, Y., Lysyanskaya, A.: ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In: ACM Computer and Communications Security—CCS '04. (2004) 354–363
12. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Advances in Cryptology—CRYPTO '05. (2005) 258–275
13. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: ACM Computer and Communications Security—CCS '06. (2006) 211–220
14. Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with short ciphertexts). In: Advances in Cryptology—EUROCRYPT '09. (2009) 171–188
15. Barth, A., Boneh, D., Waters, B.: Privacy in encrypted content distribution using private broadcast encryption. In: Financial Cryptography—FC '06. (2006) 52–64
16. Libert, B., Paterson, K.G., Quaglia, E.A.: Anonymous broadcast encryption. In: IACR Public Key Cryptography—PKC '12. (2012) 206–224

17. Krzywiecki, L., Kubiak, P., Kutylowski, M.: A revocation scheme preserving privacy. In: Information Security and Cryptology—Inscrypt '06. (2006) 130–143
18. Yu, S., Ren, K., Lou, W.: Attribute-based on-demand multicast group setup with receiver anonymity. In: Security and Privacy in Communication Networks—SecureComm '08. (2008) 18:1–18:6
19. Jarecki, S., Liu, X.: Unlinkable secret handshakes and key-private group key management schemes. In: Applied Cryptography and Network Security—ACNS '07. (2007) 270–287
20. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to Anonymous IBE, and extensions. In: Advances in Cryptology—CRYPTO '05. (2005) 205–222
21. Gentry, C.: Practical identity-based encryption without random oracles. In: Advances in Cryptology—EUROCRYPT '06. (2006) 445–464
22. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Advances in Cryptology—CRYPTO '84. (1984) 47–53
23. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Advances in Cryptology—CRYPTO '01. (2001) 213–229
24. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Advances in Cryptology—CRYPTO '04. (2004) 443–459
25. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: IEEE Symposium on Foundations of Computer Science—FOCS '07. (2007) 647–657
26. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Advances in Cryptology—CRYPTO '09. (2009) 619–636
27. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Advances in Cryptology—CRYPTO '06. (2006) 290–307
28. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Theory of Cryptography—TCC '10. (2010) 480–497
29. Cash, D., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. In: Advances in Cryptology—EUROCRYPT '08. (2008) 127–145

A Security Proofs

For ease of reference, we report below some notation that will be used in the proofs presented in this section.

Notation. $U = \{1, \dots, N\}$ is the universe of users. $\mathcal{T}$ denotes the binary tree of N users in the system with respect to the CS method. Let r be the number of revoked users and $L = \left\lfloor r \log\left(\frac{N}{r}\right)\right\rfloor$. For $b \in \{0, 1\}$, let S_b be the set of authorized receivers chosen by the adversary in the challenge phase ($|S_0| = |S_1|$). Cover_b is the family of subtrees covering the set S_b according to the CS method. Let $l_b = |\mathsf{Cover}_b|$. For each subtree T_j^b in Cover_b, let HID_j^b be the hierarchical identifier associated with the root of T_j^b where $1 \leq j \leq l_b$.

A.1 Proof of Theorem 1

Theorem 1. If $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ is (t, q_{sk}, ϵ)-$\mathsf{AIBE\text{-}IND\text{-}CPA}$ secure, then the above construction is $\left(t, q_{sk}, 2\,\epsilon\, r \log\left(\frac{N}{r}\right)\right)$-$\mathsf{oABE\text{-}IND\text{-}CPA}$ secure.

Proof. We organize our proof as a sequence of games, $\mathrm{Game}_0^0, \dots, \mathrm{Game}_{l_0}^0 \equiv \mathrm{Game}_{l_1}^1, \dots, \mathrm{Game}_0^1$, between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. In the first game (Game_0^0), $\mathcal{A}$ receives an encryption of m_0 for S_0 and in the last game (Game_0^1), $\mathcal{A}$ receives an encryption of m_1 for S_1.

Game_0^0: corresponds to the game given in Definition 5 when the challenge bit b is fixed to 0. The interaction between $\mathcal{A}$ and $\mathcal{C}$ during *Setup*, *Phase 1*, and *Phase 2* follow exactly as specified in the construction Π given in Sect. 4.1. During *Challenge*, $\mathcal{A}$ gives $\mathcal{C}$ two equal length messages $m_0, m_1 \in \mathcal{MSP}$ and two equal length sets of user identities $S_0, S_1 \subseteq U$ with the restriction that $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$, where Rev is the set of users that $\mathcal{A}$ corrupted during *Phase 1*. $\mathcal{C}$ computes the challenge ciphertext c^*, which will subsequently be sent to $\mathcal{A}$, as follows,

1. For $j = 1$ to l_0, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^0_j, m_0)$.
2. Choose $\widetilde{m} \xleftarrow{\$} \mathcal{MSP}$.
3. For $j = l_0 + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \big(c_{\pi(1)}, \ldots, c_{\pi(L)}\big)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.

Eventually, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

Game^0_h $(1 \le h \le l_0)$**:** is similar to Game$^0_{h-1}$, but $\mathcal{C}$ computes the challenge ciphertext c^* as follows,

1. For $j = 1$ to $l_0 - h$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^0_j, m_0)$.
2. Choose $\widetilde{m} \xleftarrow{\$} \mathcal{MSP}$.
3. For $j = l_0 - h + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \big(c_{\pi(1)}, \ldots, c_{\pi(L)}\big)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.

At the end, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

Game$^1_{l_1}$: is identical to Game$^0_{l_0}$

Game^1_k $(0 \le k < l_1)$**:** is similar to Game$^1_{k+1}$, but the challenge ciphertext c^* is now computed by $\mathcal{C}$ as,

1. For $j = 1$ to $l_1 - k$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^1_j, m_1)$.
2. Choose $\widetilde{m} \xleftarrow{\$} \mathcal{MSP}$.
3. For $j = l_1 - k + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \big(c_{\pi(1)}, \ldots, c_{\pi(L)}\big)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.

Finally, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

For $0 \le i \le l_0$ and $0 \le j \le l_1$, let $\mathsf{Adv}^{0,i}_{\mathcal{A},\Pi}$ and $\mathsf{Adv}^{1,j}_{\mathcal{A},\Pi}$ denote $\mathcal{A}$'s advantage of winning Game^0_i and Game^1_j respectively. In Lemma 1, we show that if the underlying AIBE scheme is (t, q_{sk}, ϵ)-AIBE-IND-CPA secure, then $\mathcal{A}$'s advantage of distinguishing Game$^0_{h-1}$ from Game^0_h is at most ϵ. Similarly, Lemma 2 states that under similar conditions $\mathcal{A}$'s advantage of distinguishing Game$^1_{k+1}$ from Game^1_k is at most ϵ. Therefore, we have,

$$\begin{aligned}\left|\mathsf{Adv}^{0,0}_{\mathcal{A},\Pi} - \mathsf{Adv}^{1,0}_{\mathcal{A},\Pi}\right| &\le \epsilon\,(l_0 + l_1)\\ &\le 2\,\epsilon\, L\\ &\le 2\,\epsilon\, r \log\left(\frac{N}{r}\right).\end{aligned}$$

Lemma 1. *For $1 \le h \le l_0$, if the underlying* AIBE *scheme Π' is (t, q_{sk}, ϵ)-*AIBE-IND-CPA *secure, then $\mathcal{A}$'s adv. of distinguishing Game$^0_{h-1}$ from Game^0_h is at most ϵ:*

$$\left|\mathsf{Adv}^{0,h-1}_{\mathcal{A},\Pi} - \mathsf{Adv}^{0,h}_{\mathcal{A},\Pi}\right| \le \epsilon.$$

Proof. We build a PPT adversary $\mathcal{B}$ that runs the AIBE-IND-CPA game with its challenger $\mathcal{C}'$ as follows. First, $\mathcal{B}$ receives the public key PK' of the AIBE scheme from $\mathcal{C}'$. Next, $\mathcal{B}$ internally executes the oABE-IND-CPA game with $\mathcal{A}$ in order to gain advantage in the AIBE-IND-CPA game. The specifics of the interaction between $\mathcal{C}'$, $\mathcal{B}$, and $\mathcal{A}$ are given below.

Setup: $\mathcal{B}$ forwards PK' to $\mathcal{A}$. $\mathcal{B}$ also initializes the set of revoked users Rev to be empty.

Phase 1: When $\mathcal{A}$ invokes a secret key query for user i, first, $\mathcal{B}$ computes HID_i, which is the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. Next, for $j = 1$ to $n+1$, $\mathcal{B}$ obtains the secret key $sk_{i,j}$ of the identity $\mathsf{HID}_{i|j}$ from its challenger $\mathcal{C}'$. After adding i to Rev, $\mathcal{B}$ sends to $\mathcal{A}$ the secret key of the user i as $sk_i = (sk_{i,1}, \ldots, sk_{i,n+1})$.

Challenge: $\mathcal{B}$ receives from $\mathcal{A}$ two equal length messages $m_0, m_1 \in \mathcal{MSP}$ and two equal length sets of user identities $S_0, S_1 \subseteq U$ with the restriction that $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$. $\mathcal{B}$ draws $\widetilde{m} \overset{\$}{\leftarrow} \mathcal{MSP}$ and computes the components of its challenge query as follows,

$$id'_0 = \mathsf{HID}^0_{l_0-h+1}, \quad id'_1 = \mathtt{dummy} \qquad m'_0 = m_0, \quad m'_1 = \widetilde{m}$$

Observe that the condition $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$, together with the key assignment strategy of the CS method guarantees that the identity id'_0 hadn't been queried to $\mathcal{B}$'s extraction oracle, and thus this is a valid challenge query to $\mathcal{C}'$.

$\mathcal{B}$ sends the two identities id'_0, id'_1 and the two messages m'_0, m'_1 as the challenge query to $\mathcal{C}'$. $\mathcal{C}'$ picks a random bit $b \in \{0, 1\}$ and sends $c^{*\prime} \leftarrow \mathsf{Enc}(\mathrm{PK}', id'_b, m'_b)$ to $\mathcal{B}$.

Finally, $\mathcal{B}$ computes the challenge ciphertext c^*, which is eventually sent to $\mathcal{A}$, as follows,

1. For $j = 1$ to $l_0 - h$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^0_j, m_0)$.
2. Set $c_{l_0-h+1} = c^{*\prime}$.
3. For $j = l_0 - h + 2$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \big(c_{\pi(1)}, \dots, c_{\pi(L)}\big)$, where $\pi : \{1, \dots, L\} \to \{1, \dots, L\}$ is a random permutation.

Phase 2: This phase is handled similarly to *Phase 1* with the usual restriction that $\mathcal{A}$ does *not* invoke a secret key query i such that $i \in S_0 \cup S_1$.

Guess: $\mathcal{A}$ outputs a guess b' and $\mathcal{B}$ passes this bit as its guess for b to $\mathcal{C}'$.

Observe that, by construction, it holds that if $\mathcal{C}'$ chooses $b = 0$, then $\mathcal{B}$ is playing Game^0_{h-1}, whereas if $b = 1$, then $\mathcal{B}$ is playing Game^0_h. Therefore, $\mathcal{B}$'s $\mathsf{AIBE\text{-}IND\text{-}CPA}$ advantage is equivalent to $\mathcal{A}$'s advantage in distinguishing Game^0_{h-1} from Game^0_h. More formally,

$$\left|\mathsf{Adv}^{0,h-1}_{\mathcal{A},\Pi} - \mathsf{Adv}^{0,h}_{\mathcal{A},\Pi}\right| = \mathsf{Adv}^{\mathsf{AIBE\text{-}IND\text{-}CPA}}_{\mathcal{B},\Pi'} \leq \epsilon.$$

Lemma 2. *For $0 \leq k < l_1$, if the underlying* AIBE *scheme Π' is (t, q_{sk}, ϵ)-*$\mathsf{AIBE\text{-}IND\text{-}CPA}$ *secure, then $\mathcal{A}$'s adv. of distinguishing $Game^1_{k+1}$ from $Game^1_k$ is at most ϵ. More precisely,*

$$\left|\mathsf{Adv}^{1,k+1}_{\mathcal{A},\Pi} - \mathsf{Adv}^{1,k}_{\mathcal{A},\Pi}\right| \leq \epsilon.$$

Proof. The argument is analogous to the proof of Lemma 1, and is therefore omitted.

A.2 Proof of Theorem 2

Theorem 2. If $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is (t, ϵ_1)-strongly existentially unforgeable and $\Pi' = (\mathsf{Init}$, Ext, Enc, $\mathsf{Dec})$ is $(t, q_{sk}, q_d, \epsilon_2)$-$\mathsf{AIBE\text{-}IND\text{-}CCA}$ secure, then the above construction is $(t$, q_{sk}, q_d, $2(\epsilon_1 + \epsilon_2)\, r \log\left(\frac{N}{r}\right))$-$\mathsf{oABE\text{-}IND\text{-}CCA}$ secure.

Proof. We organize our proof as a sequence of games, $\mathrm{Game}^0_0, \dots, \mathrm{Game}^0_{l_0} \equiv \mathrm{Game}^1_{l_1}, \dots,$ Game^1_0, between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. In the first game (Game^0_0), $\mathcal{A}$ receives an encryption of m_0 for S_0 and in the last game (Game^1_0), $\mathcal{A}$ receives an encryption of m_1 for S_1.

Game^0_0: corresponds to the game given in Definition 4 when the challenge bit b is fixed to 0. The interaction between $\mathcal{A}$ and $\mathcal{C}$ during *Setup*, *Phase 1*, and *Phase 2* follow exactly as specified in the construction Π given in Sect. 4.2. During *Challenge*, $\mathcal{A}$ gives $\mathcal{C}$ two equal length messages $m_0, m_1 \in \mathcal{MSP}$ and two equal length sets of user identities $S_0, S_1 \subseteq U$ with the restriction that $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$, where Rev is the set of users that $\mathcal{A}$ corrupted during *Phase 1*. $\mathcal{C}$ computes the challenge ciphertext C^*, which will subsequently be sent to $\mathcal{A}$, as follows,

1. Generate $(\mathrm{VK}^*, \mathrm{SK}^*) \leftarrow \mathsf{Sig\text{-}Gen}(1^\lambda)$.
2. For $j = 1$ to l_0, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^0_j, \mathrm{VK}^*||m_0)$.
3. Choose a random string $\widetilde{m}$ of the same length as $\mathrm{VK}^*||m_0$.
4. For $j = l_0 + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
5. Set $c^* = \big(c_{\pi(1)}, \ldots, c_{\pi(L)}\big)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.
6. Generate $\sigma^* \leftarrow \mathsf{Sign}(\mathrm{SK}^*, \mathrm{VK}^*||c^*)$, and output $C^* = \sigma^*||c^*$.

Eventually, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

Game^0_h $(1 \le h \le l_0)$**:** is similar to Game$^0_{h-1}$, but $\mathcal{C}$ computes the challenge ciphertext c^* as follows,

1. Generate $(\mathrm{VK}^*, \mathrm{SK}^*) \leftarrow \mathsf{Sig\text{-}Gen}(1^\lambda)$.
2. For $j = 1$ to $l_0 - h$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^0_j, \mathrm{VK}^*||m_0)$.
3. Choose a random string $\widetilde{m}$ of the same length a $\mathrm{VK}^*||m_0$.
4. For $j = l_0 - h + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
5. Set $c^* = \big(c_{\pi(1)}, \ldots, c_{\pi(L)}\big)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.
6. Generate $\sigma^* \leftarrow \mathsf{Sign}(\mathrm{SK}^*, \mathrm{VK}^*||c^*)$, and output $C^* = \sigma^*||c^*$.

At the end, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

Game$^1_{l_1}$: is identical to Game$^0_{l_0}$

Game^1_k $(0 \le k < l_1)$**:** is similar to Game$^1_{k+1}$, but the challenge ciphertext c^* is now computed by $\mathcal{C}$ as,

1. Generate $(\mathrm{VK}^*, \mathrm{SK}^*) \leftarrow \mathsf{Sig\text{-}Gen}(1^\lambda)$.
2. For $j = 1$ to $l_1 - k$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^1_j, \mathrm{VK}^*||m_1)$.
3. Choose a random string $\widetilde{m}$ of the same length as $\mathrm{VK}^*||m_1$.
4. For $j = l_1 - k + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
5. Set $c^* = \big(c_{\pi(1)}, \ldots, c_{\pi(L)}\big)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.
6. Generate $\sigma^* \leftarrow \mathsf{Sign}(\mathrm{SK}^*, \mathrm{VK}^*||c^*)$, and output $C^* = \sigma^*||c^*$.

Finally, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

For $0 \le i \le l_0$ and $0 \le j \le l_1$, let $\mathsf{Adv}^{0,i}_{\mathcal{A},\Pi}$ and $\mathsf{Adv}^{1,j}_{\mathcal{A},\Pi}$ denote $\mathcal{A}$'s advantage of winning Game^0_i and Game^1_j respectively. In Lemma 3, we show that if the underlying one-time signature scheme and AIBE scheme are respectively (t, ϵ_1)-strongly unforgeable and (t, q_{sk}, ϵ_2)-$\mathsf{AIBE\text{-}IND\text{-}CCA}$ secure, then $\mathcal{A}$'s advantage of distinguishing Game$^0_{h-1}$ from Game^0_h is at most $\epsilon_1 + \epsilon_2$. Similarly, Lemma 4 states that under analogous conditions $\mathcal{A}$'s advantage of distinguishing Game$^1_{k+1}$ from Game^1_k is again at most $\epsilon_1 + \epsilon_2$. Therefore, we have,

$$\begin{aligned}\left|\mathsf{Adv}^{0,0}_{\mathcal{A},\Pi} - \mathsf{Adv}^{1,0}_{\mathcal{A},\Pi}\right| &\le (\epsilon_1 + \epsilon_2)\,(l_0 + l_1)\\ &\le 2\,(\epsilon_1 + \epsilon_2)\,L\\ &\le 2\,(\epsilon_1 + \epsilon_2)\,r \log\left(\frac{N}{r}\right).\end{aligned}$$

Lemma 3. *For $1 \le h \le l_0$, if the underlying one-time signature scheme Σ is (t, ϵ_1)-strongly unforgeable and the* AIBE *scheme Π' is (t, q_{sk}, ϵ_2)-*$\mathsf{AIBE\text{-}IND\text{-}CCA}$ *secure, then $\mathcal{A}$'s adv. of distinguishing Game$^0_{h-1}$ from Game^0_h is at most $\epsilon_1 + \epsilon_2$:*

$$\left|\mathsf{Adv}^{0,h-1}_{\mathcal{A},\Pi} - \mathsf{Adv}^{0,h}_{\mathcal{A},\Pi}\right| \le (\epsilon_1 + \epsilon_2).$$

Proof. We build a PPT adversary $\mathcal{B}$ that runs the $\mathsf{AIBE\text{-}IND\text{-}CCA}$ game with its challenger $\mathcal{C}'$ as follows. First, $\mathcal{B}$ receives the public key PK' of the AIBE scheme from $\mathcal{C}'$. Next, $\mathcal{B}$ internally executes the $\mathsf{oABE\text{-}IND\text{-}CCA}$ game with $\mathcal{A}$ in order to gain advantage in the $\mathsf{AIBE\text{-}IND\text{-}CCA}$ game. The specifics of the interaction between $\mathcal{C}'$, $\mathcal{B}$, and $\mathcal{A}$ are given below.

Setup: $\mathcal{B}$ forwards PK' to $\mathcal{A}$. $\mathcal{B}$ also initializes the set of revoked users Rev to be empty.

Phase 1: When $\mathcal{A}$ invokes a secret key query for user i, first, $\mathcal{B}$ computes HID_i, which is the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. Next, for $j = 1$ to $n+1$, $\mathcal{B}$ obtains the secret key $sk_{i,j}$ of the identity $\mathsf{HID}_{i|j}$ from its challenger $\mathcal{C}'$. After adding i to Rev, $\mathcal{B}$ sends to $\mathcal{A}$ the secret key of the user i as $sk_i = (sk_{i,1}, \ldots, sk_{i,n+1})$.

When $\mathcal{A}$ invokes a decryption query $(i, C = \sigma||(c_1, \ldots, c_L))$, $\mathcal{B}$ computes HID_i, and for each $j = 1$ to $n+1$, $\mathcal{B}$ proceeds as follows:

a. If $\mathcal{B}$ obtained the secret key $sk_{i,j}$ corresponding to the identity $\mathsf{HID}_{i|j}$ in the process of responding to a previous secret key query, then $\mathcal{B}$ attempts to decrypt in turn all ciphertext components $c_1, \ldots, c_L$ in C using the secret key $sk_{i,j}$. If any of these decryption attempts yield a non-$\perp$ value $\mathrm{VK}||m$, then $\mathcal{B}$ returns m to $\mathcal{A}$ if $\mathsf{Vrfy}(\mathrm{VK}, \sigma, \mathrm{VK}||c)$, where $c = (c_1, \ldots, c_L)$. Otherwise, $\mathcal{B}$ continues to next j.

b. If $\mathcal{B}$ did not obtain the secret key $sk_{i,j}$ of the identity $\mathsf{HID}_{i|j}$ from an earlier secret key query, then $\mathcal{B}$ makes L decryption queries to its challenger $\mathcal{C}'$, one for each ciphertext component $c_1, \ldots, c_L$, all under identity $\mathsf{HID}_{i|j}$. If any of these decryption queries return a non-$\perp$ value $\mathrm{VK}||m$, then $\mathcal{B}$ returns m to $\mathcal{A}$ if $\mathsf{Vrfy}(\mathrm{VK}, \sigma, \mathrm{VK}||c)$. Otherwise, $\mathcal{B}$ continues to next j.

If all the above decryption attempts return $\perp$, then $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.

Challenge: $\mathcal{B}$ receives from $\mathcal{A}$ two equal length messages $m_0, m_1 \in \mathcal{MSP}$ and two equal length sets of user identities $S_0, S_1 \subseteq U$ with the restriction that $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$. $\mathcal{B}$ generates $(\mathrm{VK}^*, \mathrm{SK}^*) \leftarrow \mathsf{Sig\text{-}Gen}(1^\lambda)$, selects a random string $\widetilde{m}$ of the same length as $\mathrm{VK}^*||m_0$, and sets:

$$id'_0 = \mathsf{HID}^0_{l_0-h+1}, \quad id'_1 = \mathtt{dummy} \qquad m'_0 = \mathrm{VK}^*||m_0, \quad m'_1 = \widetilde{m}$$

Next, $\mathcal{B}$ sends the two identities id'_0, id'_1 and the two messages m'_0, m'_1 as the challenge query to $\mathcal{C}'$. $\mathcal{C}'$ picks a random bit $b \in \{0,1\}$ and responds to $\mathcal{B}$ with $c^{*\prime} \leftarrow \mathsf{Enc}(\mathrm{PK}', id'_b, m'_b)$.

Finally, $\mathcal{B}$ computes the challenge ciphertext C^*, which is eventually sent to $\mathcal{A}$, as follows,

1. For $j = 1$ to $l_0 - h$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^0_j, \mathrm{VK}^*||m_0)$.
2. Set $c_{l_0-h+1} = c^{*\prime}$.
3. For $j = l_0 - h + 2$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \big(c_{\pi(1)}, \ldots, c_{\pi(L)}\big)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.
5. Generate $\sigma^* \leftarrow \mathsf{Sign}(\mathrm{SK}^*, \mathrm{VK}^*||c^*)$, and set $C^* = \sigma^*||c^*$.

Phase 2: Secret key queries are handled similarly to *Phase 1*, with the usual restriction that $\mathcal{A}$ does *not* invoke a secret key query i such that $i \in S_0 \cup S_1$.

As for decryption queries, $\mathcal{B}$ replies to $(i, C = \sigma||(c_1, \ldots, c_L))$, according to one of the following cases:

– If $C = C^*$ and $i \notin S_0 \cup S_1$, then $\mathcal{B}$ proceeds as in *Phase 1.* (Note that in this case $\mathcal{B}$'s output will be $\perp$, as it should be.)

– If $C = C^*$, and $i \in S_0 \cup S_1$, $\mathcal{B}$ just rejects, since $\mathcal{A}$ is submitting an invalid query.

– If $C \neq C^*$ and $i \notin S_0$, then $\mathcal{B}$ proceeds as in *Phase 1.*

– If $C \neq C^*$ and $i \in S_0$, then $\mathcal{B}$ computes HID_i, and proceeds as follows:

- If for all $j = 1$ to $n+1$, it is the case that $\mathsf{HID}_{i|j} \neq \mathsf{HID}^0_{l_0-h+1}$, then $\mathcal{B}$ proceeds as in *Phase 1* (Case b.). Observe that the condition $\forall j \in [1, n+1]$ $(\mathsf{HID}_{i|j} \neq \mathsf{HID}^0_{l_0-h+1})$ ensures that all the decryption queries that $\mathcal{B}$ will make to its challenger $\mathcal{C}'$ in the process of responding to $\mathcal{A}$'s queries are allowable.
- If $\exists\ j \in [1, n+1]$ such that $\mathsf{HID}_{i|j} = \mathsf{HID}^0_{l_0-h+1}$, and $c^{*\prime}$ does not appear among the ciphertext components of C, then again $\mathcal{B}$ proceeds as in *Phase 1* (Case b.). Observe that the condition that C does not contain $c^{*\prime}$ ensures that also in this case all the decryption queries that $\mathcal{B}$ will make to its challenger $\mathcal{C}'$ in the process of responding to $\mathcal{A}$'s queries are allowable.
- If $\exists\ j \in [1, n+1]$ such that $\mathsf{HID}_{i|j} = \mathsf{HID}^0_{l_0-h+1}$, but $c^{*\prime}$ appears among the ciphertext components of C, then $\mathcal{B}$ outputs $\perp$. To see that $\perp$ is the correct reply, observe that in the real oABE-IND-CCA game, a decryption query (i, C) of this type will trigger decryption of the $c^{*\prime}$ component. Since by construction $c^{*\prime}$ is the encryption of $\mathrm{VK}^*||m_0$, and $C \neq C^*$, by the unforgeability of the underlying one-time signature scheme, the verification test of Step 1b. of the decryption algorithm would fail, thus yielding $\perp$ as output.

Guess: $\mathcal{A}$ outputs a guess b' and $\mathcal{B}$ passes this bit as its guess for b to $\mathcal{C}'$.

Observe that, by construction, it holds that if $\mathcal{C}'$ chooses $b = 0$, then $\mathcal{B}$ is playing Game^0_{h-1}, whereas if $b = 1$, then $\mathcal{B}$ is playing Game^0_h. Therefore, up to forgeries of the underlying one-time signature scheme, $\mathcal{B}$'s AIBE-IND-CCA advantage is essentially $\mathcal{A}$'s advantage in distinguishing Game^0_{h-1} from Game^0_h:

$$\left|\mathsf{Adv}^{0,h-1}_{\mathcal{A},\Pi} - \mathsf{Adv}^{0,h}_{\mathcal{A},\Pi}\right| \leq (\epsilon_1 + \epsilon_2).$$

Lemma 4. *For $0 \leq k < l_1$, if the underlying one-time signature scheme Σ is (t, ϵ_1)-strongly unforgeable and the underlying* AIBE *scheme Π' is (t, q_{sk}, ϵ_2)-*AIBE-IND-CCA *secure, then $\mathcal{A}$'s adv. of distinguishing Game$^1_{k+1}$ from Game^1_k is at most ϵ. More precisely,*

$$\left|\mathsf{Adv}^{1,k+1}_{\mathcal{A},\Pi} - \mathsf{Adv}^{1,k}_{\mathcal{A},\Pi}\right| \leq (\epsilon_1 + \epsilon_2).$$

Proof. The argument is analogous to the proof of Lemma 3, and is therefore omitted.