Bounded-Collusion Attribute-Based Encryption from Minimal Assumptions

Gene Itkis[1], Emily Shen[1], Mayank Varia[2], David Wilson[1], and Arkady Yerukhimovich[1(✉)]

[1] MIT Lincoln Laboratory, Lexington, MA, USA
{itkis,emily.shen,david.wilson,arkady}@ll.mit.edu
[2] Boston University, Boston, MA, USA
varia@bu.edu

Abstract. Attribute-based encryption (ABE) enables encryption of messages under access policies so that only users with attributes satisfying the policy can decrypt the ciphertext. In standard ABE, an arbitrary number of colluding users, each without an authorized attribute set, cannot decrypt the ciphertext. However, all existing ABE schemes rely on concrete cryptographic assumptions such as the hardness of certain problems over bilinear maps or integer lattices. Furthermore, it is known that ABE cannot be constructed from generic assumptions such as public-key encryption using black-box techniques.

In this work, we revisit the problem of constructing ABE that tolerates collusions of arbitrary but *a priori* bounded size. We present two ABE schemes secure against bounded collusions that require only semantically secure public-key encryption. Our schemes achieve significant improvement in the size of the public parameters, secret keys, and ciphertexts over the previous construction of bounded-collusion ABE from minimal assumptions by Gorbunov et al. (CRYPTO 2012). In fact, in our second scheme, the size of ABE secret keys does not grow at all with the collusion bound. As a building block, we introduce a multidimensional secret-sharing scheme that may be of independent interest. We also obtain bounded-collusion symmetric-key ABE (which requires the secret key for encryption) by replacing the public-key encryption with symmetric-key encryption, which can be built from the minimal assumption of one-way functions.

Keywords: Attribute-based encryption · Public-key encryption · Bounded collusion · Secret sharing

G. Itkis, E. Shen, D. Wilson and A. Yerukhimovich—This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering (ASDR&E) under Air Force Contract No. FA8721-05-C-0002. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government.
M. Varia—This material is based upon work supported by the National Science Foundation under Grant No. 1414119.

S. Fehr (Ed.): PKC 2017, Part II, LNCS 10175, pp. 67–87, 2017.
DOI: 10.1007/978-3-662-54388-7_3

1 Introduction

In traditional public-key encryption, data is encrypted for an individual user whose public key is known at the time of encryption, and only the target user is able to decrypt the resulting ciphertext. However, many applications require encryption with more expressive access control capabilities. For example, electronic medical records contain a wealth of sensitive patient information that should be accessible only to medical administrators (e.g., doctors, nurses, pharmacists, and researchers) whose credentials satisfy complex access policies based on their roles and relationships to the patient [2].

For these applications, straightforward encryption solutions are inadequate for two reasons. First, the ciphertext must be decryptable by potentially many users with distinct keys. The trivial solution of encrypting the data separately to each user results in long ciphertexts. A long line of work on broadcast encryption (e.g., [5,11,23]) aims to reduce the ciphertext size for this problem. Second, the identities of the authorized users may not be known to the encryptor; instead of encrypting to individual users we wish to encrypt to access policies so that only users whose credentials satisfy the policy can decrypt. The trivial solution of providing a separate key for each group of attributes results in long keys for the recipients of messages.

Attribute-based encryption (ABE), introduced by Sahai and Waters [26], addresses both of these issues. In ABE, each secret key corresponds to a predicate f, and each ciphertext corresponds to a message and an index ind. Decryption returns the message if and only if $f(\mathsf{ind}) = 1$. Thus, ABE allows automatic enforcement of any access policy that can be expressed as the evaluation of $f(\mathsf{ind})$. Two commonly considered special cases of ABE are ciphertext-policy ABE (CP-ABE) [4], where the secret key predicate is a set of attributes and the ciphertext index is an access policy over attributes, and key-policy ABE (KP-ABE) [18], where the roles of the index and the predicate are reversed.

Since the introduction of ABE, many constructions and related primitives have appeared in the literature (e.g., [4,12,16,18,22,24,28]). ABE has also been implemented in some applications, including the protection of electronic medical records [2]; we refer readers to [19, Sect. 3.2] for a longer overview of the history of ABE.

However, all known constructions of ABE rely on concrete assumptions such as the hardness of certain problems over bilinear maps or integer lattices rather than generic assumptions such as the existence of CPA-secure public-key encryption. In fact, it is known that, when using black-box techniques, the security of ABE cannot be based on such generic assumptions [6,21].

The difficulty of building ABE from generic assumptions stems from its *collusion resistance* requirement, which states that two or more users, neither of whose attributes satisfy the policy embedded in a ciphertext, should not be able to decrypt the message using their joint key material. Intuitively, for CP-ABE this requires the secret key corresponding to a set of attributes to be "bound" together so that the contribution that each attribute makes to the key cannot be detached and re-purposed toward decrypting a message requiring a different combination of attributes. ABE typically requires security against *unbounded*

collusion. That is, even if a very large and *a priori* unbounded number of users collude, they should fail to decrypt any ciphertexts that none of them can decrypt individually.

In this work, we consider a relaxation of the unbounded collusion requirement and instead consider schemes that are secure against an *a priori* bounded number of colluders. Positive results have recently been shown in constructing *bounded-collusion ABE* (BC-ABE) schemes assuming only the existence of public-key encryption [15,25].[1] We stress that this relaxation does not limit the number of keys that may be issued, but rather only the number of colluders that the scheme can withstand.

Such generic constructions of ABE based on public-key encryption have several benefits. First, they can be instantiated from a number of standard cryptographic hardness assumptions. Second, by replacing CPA-secure public-key encryption with its symmetric-key counterpart, these schemes directly yield a construction of symmetric-key ABE schemes that require the secret key for encryption as well.[2] In particular, this implies that bounded-collusion symmetric-key ABE can be constructed from the minimal assumption of the existence of one-way functions. By contrast, constructions of ABE based on specific assumptions lack a clear transformation into symmetric-key ABE without still relying on "public-key" assumptions.

However, the only known constructions of BC-ABE from public-key encryption [15] require keys and ciphertexts that grow very quickly with the collusion bound (see Table 1). Thus, it remains worthwhile to reduce the key and ciphertext length in constructions of bounded-collusion ABE to understand what can be achieved using these minimal assumptions.

1.1 Our Results

In this paper we address exactly this problem, showing two different constructions of bounded-collusion ABE based only on the existence of public-key encryption, achieving shorter key sizes, public parameters, and ciphertexts. We adopt the two-step procedure taken by Gorbunov et al. [15]: first design an ABE scheme that is secure against an adversary with only a single key (which we call a *1-ABE scheme*), and then design a bootstrapping procedure that yields a BC-ABE scheme secure against a larger number of collusions q (which we call a *q-ABE scheme*). Indeed, we retain the 1-ABE scheme of [15], which can be instantiated based only on CPA-secure public-key encryption. Therefore, the focus of our work is to reduce the dependence on q in the construction of q-ABE from 1-ABE. Specifically, we show a construction satisfying the following theorem:

[1] These works actually build bounded-collusion functional encryption (FE), a stronger primitive that implies ABE. The bounded-collusion FE construction [15] actually requires an additional assumption of the existence of bounded-degree PRGs, but, as the authors show, this assumption is not needed for bounded-collusion ABE. For the purposes of this paper, we will only discuss the ABE constructions.

[2] Symmetric-key ABE is useful for applications such as publish-subscribe allowing a single publisher to disseminate information to subscribers based on their attributes or interests.

Theorem 1 (Informal). *Suppose there exists a public-key (resp., symmetric-key) 1-ABE scheme for a class of access policies. Then there exists a public-key (resp., symmetric-key) BC-ABE scheme for the same class of access policies tolerating collusions of size at most q with the following characteristics: public parameters consisting of $O(\frac{q^2}{\log q}\lambda)$ 1-ABE encryption keys, secret keys consisting of $O(\frac{1}{\log q}\lambda)$ 1-ABE keys, and ciphertexts consisting of $O(\frac{q^2}{\log q}\lambda)$ 1-ABE ciphertexts, where λ is the security parameter.*

We formalize and prove this theorem in Sect. 5. We then instantiate the 1-ABE scheme with the construction of Sahai and Seyalioglu [25] (subsequently improved to handle full, adaptive security by Gorbunov et al. [15]), which gives 1-ABE for the access policies expressed by arbitrary Boolean circuits from CPA-secure encryption. This immediately yields the following result:

Corollary 1. *If public-key (respectively, symmetric-key) encryption exists, then there exist public-key (resp., symmetric-key) ABE schemes for access policies expressed by boolean circuits tolerating collusion of size at most q. The sizes of the public parameters, secret keys, and ciphertexts in the resulting BC-ABE scheme come from two sources: (1) the use of CPA-secure encryption to construct 1-ABE (e.g., in [15, 25]) and (2) the use of 1-ABE to construct q-ABE in Theorem 1. In particular, the only dependencies of these parameters on q come from Theorem 1, since any 1-ABE construction from CPA-secure encryption is clearly independent of q.*

1.2 Comparison to Prior Work

We construct two schemes in this paper: a basic scheme in Sect. 4 that is easier to analyze but whose bounds are slightly weaker than those in Theorem 1, and then an improved scheme that fully meets the theorem. This section and Table 1 compare the parameters of our schemes with two related works: Dodis et al.'s bounded-collusion identity-based encryption (IBE) scheme [10] and Gorbunov et al.'s bounded-collusion ABE scheme [15].

Our basic scheme has asymptotic dependence on q that is roughly comparable to the Dodis et al. [10] construction of bounded-collusion IBE, a weaker primitive than ABE, from public-key encryption, while avoiding the need for cover-free sets used by that construction. Specifically, our scheme has shorter secret keys but larger ciphertexts; the asymptotic size of the public parameters is the same in both constructions.

Our basic scheme is also a significant improvement over the bounded-collusion ABE scheme of [15], in which both the public parameters and the ciphertext grow as $O(q^4)$. Indeed, the secret key size in our basic scheme does not grow with the collusion bound. This is a significant improvement allowing us to keep secret key sizes short even when tolerating a high collusion bound. Also, the dependence on q of the ciphertext size of our basic scheme matches that of the best known constructions of bounded-collusion functional encryption (which implies ABE) from lattice assumptions [1].

Table 1. Comparison of bounded-collusion ABE schemes tolerating collusions of size at most q (note: DKXY only provides IBE). Sizes are given in terms of number of 1-ABE keys or 1-ABE ciphertexts. Here λ is a security parameter.

	DKXY [10]	GVW [15]	Basic scheme	Improved scheme
Public parameters	$O(q^2\lambda)$	$O(q^4\lambda)$	$O(q^2\lambda)$	$O\left(\frac{q^2}{\log q}\lambda\right)$
Secret keys	$O(q\lambda)$	$O(q^2\lambda)$	$O(\lambda)$	$O\left(\frac{1}{\log q}\lambda\right)$
Ciphertexts	$O(q\lambda)$	$O(q^4\lambda)$	$O(q^2\lambda)$	$O\left(\frac{q^2}{\log q}\lambda\right)$

Our improved scheme further reduces the size of public parameters, secret keys, and ciphertexts each by a factor of $\log q$. This leads to the somewhat counterintuitive property that the size of secret keys *decreases* as the collusion bound increases!

1.3 Our Techniques

Our main technique follows the same high-level approach taken by Gorbunov et al. [15]. Specifically, during setup, N key pairs for a 1-ABE scheme are generated. The secret keys become the master secret key of the BC-ABE scheme while the public keys become the public parameters. Then, every BC-ABE secret key consists of a subset of the secret keys. To encrypt a message m with an index ind, the message is first secret-shared and then each share is encrypted under ind using a different 1-ABE public key. To make this work, the subset of keys included in a BC-ABE secret key and the secret sharing are chosen in such a way that if $f(\mathsf{ind}) = 1$ for the predicate f encoded in a secret key, then that key will allow the recovery of sufficiently many shares of m so decryption will succeed. However, any set of q keys not satisfying ind reveals no information about m. In particular, such a set of keys cannot be combined to recover the appropriate shares to reconstruct m.

In [15] this property is achieved by using a t-out-of-n secret sharing of the message and then partitioning the secret keys in such a way that sets of keys included in different BC-ABE secret keys have small pairwise intersections. Since at least t key intersections are needed to recover the message (each intersection allows the attacker to recover one share), this guarantees that a large number of keys is needed.

Our basic scheme improves on this technique by (1) using an n-out-of-n secret sharing of the message and encrypting each share under l independent 1-ABE keys and then (2) for each BC-ABE secret key giving 1 out of the l possible keys to recover each share to reduce the probability of key intersection. This requires an adversary to be able to reconstruct all of the n top level shares by getting enough intersections for each of them. We show that this approach allows us to reduce the size of the public parameters and the ABE secret keys while still guaranteeing resistance against q bounded-collusions with overwhelming probability.

Our improved scheme uses a *multi-dimensional secret-sharing* algorithm, which has the properties that (1) there exist small sets of shares that suffice to reconstruct the message and (2) such small sets of shares are rare, so for shares chosen at random a large number of shares is needed to reconstruct the message. By using multi-dimensional secret-sharing, the secret keys of our ABE scheme only need to include keys allowing decryption of such a small set of shares, whereas an adversary who only learns shares at random must recover a large number of shares in order to reconstruct the message. This allows us to further reduce the size of public parameters, keys, and ciphertexts by an additional logarithmic factor in the collusion bound q.

1.4 Paper Organization

The rest of the paper is organized as follows. In Sect. 2, we provide more details on related work. In Sect. 3, we give some necessary background and define bounded-collusion ABE. In Sect. 4, we present our basic construction. Then, in Sect. 5, we present our improved construction. Finally, in Sect. 6 we briefly discuss how to instantiate 1-ABE.

2 Related Work

Impossibility of Unbounded Collusion From Generic Assumptions.
Several prior works have aimed to understand the difficulty of building ABE and related primitives from generic assumptions such as CPA-secure encryption. Evidence that such constructions are unlikely was first given by Boneh et al. [6], who showed that there is no black-box construction of IBE from CPA-secure encryption or trapdoor permutations. This result was subsequently extended by Katz and Yerukhimovich [21], who also ruled out constructions of ABE for several classes of access policies. Finally, Goyal et al. [17] showed that for certain classes of access policies, ABE cannot be even constructed from the much stronger assumption that IBE exists. Note that the latter two works prove impossibility of *public-index predicate encryption*, a construct that is equivalent to ABE and that we will use in this paper as well (cf. Definition 3).

Bounded Collusion Constructions.
Our restriction to tolerating collusions of bounded size has been used before to build ABE and related primitives from (somewhat) standard assumptions. Early works [9,10] showed how to construct bounded-collusion identity-based encryption (IBE), a special case of ABE where the only formulas allowed are equalities over the set of attributes, from standard public-key encryption. Later, Goldwasser et al. [14] showed a more efficient construction of bounded-collusion IBE if the underlying encryption scheme satisfied a key-homomorphism property and had an associated hash-proof system. This latter requirement of hash-proof systems was subsequently removed by Tessaro and Wilson [27].

Going beyond IBE, Sahai and Seyalioglu [25] showed that standard public-key secure encryption can be used to achieve 1-query security for functional

encryption, a powerful generalization of ABE. This construction was then leveraged and improved by Gorbunov et al. [15] to achieve bounded-collusion security for functional encryption under the assumption that a low-depth pseudorandom number generator exists. However, their construction can be used to realize bounded-collusion ABE without this latter assumption.

Additionally, the bounded-collusion relaxation has also been used for several constructions relying on stronger computational assumptions. For example, Goldwasser et al. [13] show how to build a 1-key succinct functional encryption scheme based on any fully-homomorphic encryption and attribute-based encryption for circuits, both of which can be realized from lattice assumptions. More recently, Agrawal and Rosen [1] showed how to build a bounded-collusion functional encryption scheme achieving online/offline encryption, allowing much of the encryption procedure to be precomputed before the message is known, from a specific lattice-based functional encryption scheme for inner product functions.

3 Definitions

In this section, we provide notation and definitions of the primitives we will use.

3.1 Preliminaries

For $n \in \mathbb{N}$, we let $[n]$ denote the set of integers $\{1, \ldots, n\}$. Let negl denote a negligible function. Let PPT denote the class of algorithms that run in probabilistic polynomial time. Additionally, we assume in this work that all sets are ordered.

We first define public- and symmetric-key encryption.

Definition 1 (Encryption scheme). *A* public-key (*respectively,* symmetric-key) encryption scheme Σ *for the message space* $\mathcal{M}$ *consist of three* PPT *algorithms* KeyGen, Enc, *and* Dec *defined as follows.*

- $\mathsf{KeyGen}(1^\lambda)$ *takes as input the unary representation of the security parameter* λ *and outputs the public and private keys* $(\mathsf{pk}, \mathsf{sk})$. *(For a symmetric-key encryption scheme,* pk *must be the empty string.)*
- $\mathsf{Enc}(\mathsf{ek}, m)$ *takes as input an encryption key* ek *and a message* $m \in \mathcal{M}$ *and outputs a ciphertext* ct, *where* $\mathsf{ek} = \mathsf{pk}$ *(resp.,* $\mathsf{ek} = \mathsf{sk}$*).*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *takes as input the secret key* sk *and a ciphertext* ct *and outputs either a message* $m \in \mathcal{M}$ *or the distinguished symbol* $\perp$.

For correctness we require the following condition: for all λ *and* $m \in \mathcal{M}$, *if we compute* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{ek}, m)$, *then* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = m$.

We use a standard notion of security against chosen plaintext attacks defined in terms of a left-or-right oracle. For $b \in \{0, 1\}$, we define $\mathsf{Enc}_b(\mathsf{ek}, m_0, m_1) = \mathsf{Enc}(\mathsf{ek}, m_b)$.

Definition 2 (CPA-security for encryption). *An encryption scheme* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* CPA-secure *if for all valid* PPT *adversaries* $\mathcal{A}$*,*

$$|\Pr[\mathcal{A}^{\mathsf{Enc}_0(\mathsf{ek},\cdot,\cdot)}(1^\lambda, \mathsf{pk}) = 1] - \Pr[\mathcal{A}^{\mathsf{Enc}_1(\mathsf{ek},\cdot,\cdot)}(1^\lambda, \mathsf{pk}) = 1]| \leq \mathsf{negl}(\lambda),$$

where the probability is taken over the randomness of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*,* Enc*, and* $\mathcal{A}$*. An adversary* $\mathcal{A}$ *is* valid *if* $|m_0| = |m_1|$ *for all* Enc_b *queries* (m_0, m_1)*.*

3.2 Attribute-Based Encryption with Bounded-Collusion Security

We now define attribute-based encryption (ABE) (also called predicate encryption with public index). This definition encompasses both ciphertext-policy ABE and key-policy ABE.

Definition 3 (Attribute-based encryption scheme). *A* public-key, (*respectively,* symmetric-key) attribute-based encryption scheme Π *for a message space* $\mathcal{M}$*, an index space* $\mathcal{I}$*, and a predicate space* $\mathcal{F}$ *consists of four* PPT *algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *defined as follows.*

- $\mathsf{Setup}(1^\lambda, q)$ *takes as input the unary representation of the security parameter* λ *and (optionally) a collusion bound* q*, and outputs the master public and secret keys* $(\mathsf{MPK}, \mathsf{MSK})$*. (For a symmetric-key attribute-based encryption scheme,* MPK *must be the empty string.)*
- $\mathsf{KeyGen}(\mathsf{MSK}, f)$ *takes as input the master secret key* MSK *and a predicate* $f \in \mathcal{F}$*, and outputs a secret key* sk_f*.*
- $\mathsf{Enc}(\mathsf{EK}, m, \mathsf{ind})$ *takes as input an encryption key* EK*, a message* $m \in \mathcal{M}$*, and an index* $\mathsf{ind} \in \mathcal{I}$*, and outputs a ciphertext* ct*, where* $\mathsf{EK} = \mathsf{MPK}$ *(resp.,* $\mathsf{EK} = \mathsf{MSK}$*).*
- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$ *takes as input a secret key* sk_f *and a ciphertext* ct*, and outputs either a message* $m \in \mathcal{M}$ *or the distinguished symbol* $\perp$*.*

For correctness we require the following: for all λ*,* $q \in \mathbb{N}$*,* $m \in \mathcal{M}$*,* $\mathsf{ind} \in \mathcal{I}$*, and* $f \in \mathcal{F}$ *such that* $f(\mathsf{ind}) = 1$*, if we compute* $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, q)$*,* $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, f)$*, and* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{EK}, m, \mathsf{ind})$*, then we require* $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}) = m$*.*

We stress that in the above definition Setup takes the query bound q as a parameter; therefore, MPK and MSK may depend on q.

We now define bounded-collusion security for attribute-based encryption. Our definitions follow the functional encryption definitions of Brakerski and Segev [7]. We define security in terms of left-or-right indistinguishability. For $b \in \{0, 1\}$, we define $\mathsf{Enc}_b(\mathsf{EK}, (m_0, m_1), \mathsf{ind}) = \mathsf{Enc}(\mathsf{EK}, m_b, \mathsf{ind})$.

Definition 4 (q-query security for ABE). *An attribute-based encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* q-query secure *if for all valid* PPT *adversaries* $\mathcal{A}$ *making at most* q *key queries,*

$$\begin{aligned} Adv_{\Pi,\mathcal{A},q}(\lambda) = |\Pr[&\mathcal{A}^{\mathsf{KeyGen}(\mathsf{MSK},\cdot),\mathsf{Enc}_0(\mathsf{EK},\cdot,\cdot)}(1^\lambda, q, \mathsf{MPK}) = 1] \\ &- \Pr[\mathcal{A}^{\mathsf{KeyGen}(\mathsf{MSK},\cdot),\mathsf{Enc}_1(\mathsf{EK},\cdot,\cdot)}(1^\lambda, q, \mathsf{MPK}) = 1]| \leq \mathsf{negl}(\lambda). \end{aligned}$$

In the definition of advantage, the probabilities are taken over the randomness of $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, q)$, KeyGen, Enc, *and* $\mathcal{A}$. *An adversary* $\mathcal{A}$ *is* valid *if for all* Enc_b *queries* $((m_0, m_1), \mathsf{ind})$, $|m_0| = |m_1|$; *furthermore, if there exists any* KeyGen *query* f *such that* $f(\mathsf{ind}) = 1$, *then* $m_0 = m_1$.

4 Basic BC-ABE Construction

We now present our basic bounded-collusion construction that builds a q-query secure attribute-based encryption scheme from a 1-query secure attribute-based encryption scheme.

For intuition, consider an encryption algorithm that encrypts the message with its associated index many times under independent instances of a 1-query attribute-based encryption scheme. Let the secret key for a predicate be generated as the secret key for that predicate for one of the 1-query schemes, chosen at random. Then an authorized user (a user with a predicate satisfied by the index) can decrypt the message using the 1-query scheme for which she has a key. If two unauthorized users collude, as long as their keys are from different instances of the 1-query ABE scheme, the 1-query security property suffices to ensure that they cannot learn anything about the message.

However, this simple parallel encryption approach does not scale well. If the total number of users exceeds the number of 1-query ABE instances, there will necessarily be two users with keys from the same instance, exceeding the collusion bound for that instance.

Instead, in our construction, we first additively secret-share the message, then perform parallel encryptions as described above on each additive share. Each user is given for each additive share a key from a random 1-query ABE instance. This approach allows us to make a combinatorial argument about the number of unauthorized colluders necessary to reconstruct the message with non-negligible probability.

Note, however, that unlike the message, the index is not secret shared and is included in each of the 1-query ABE ciphertexts. For this reason our construction cannot be used to achieve q-query security for the stronger primitive of predicate encryption with private index, even if the 1-query scheme has this stronger property. Specifically, the index will be revealed any time an adversary receives two keys for any of the component 1-query schemes, thus breaking index privacy.

4.1 Construction

Let $\mathsf{1\text{-}ABE}$ be a 1-query secure attribute-based encryption scheme with message space $\mathcal{M}$, index space $\mathcal{I}$, and predicate space $\mathcal{F}$; we require that $\mathcal{M}$ have the property that the set of elements of each length form a finite group, so that we may perform additive secret sharing. Additionally, let ℓ and w be integers; we will explain later how to set these parameters based on the security parameter λ and the collusion bound q. We define the scheme $\mathsf{q\text{-}ABE}$ for message space $\mathcal{M}$,

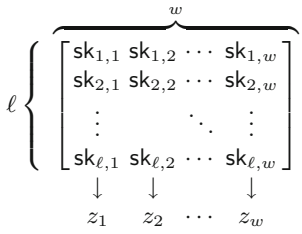

Fig. 1. Overview of our basic construction. A user with predicate $f \in \mathcal{F}$ receives w 1-ABE secret keys, one from each column, where $\mathsf{sk}_{i,j} \leftarrow \text{1-ABE.KeyGen}(\mathsf{MSK}_{i,j}, f)$. A ciphertext for a message m contains $\ell \cdot w$ 1-ABE ciphertexts, formed by using each of the ℓ keys in the j^{th} column, individually, to encrypt secret share z_j, where $m = \sum_{j=1}^{w} z_j$. One key from each column is required for decryption.

index space $\mathcal{I}$, and predicate space $\mathcal{F}$ formally below; we also refer readers to Fig. 1 for an informal visual depiction.

$\mathsf{Setup}(1^\lambda, q)$**:** For each row $i \in [\ell]$ and column $j \in [w]$, independently sample $(\mathsf{MPK}_{i,j}, \mathsf{MSK}_{i,j}) \leftarrow \text{1-ABE.Setup}(1^\lambda)$. Output $\mathsf{MPK} = \{\mathsf{MPK}_{i,j}\}_{i\in[\ell],j\in[w]}$ and $\mathsf{MSK} = \{\mathsf{MSK}_{i,j}\}_{i\in[\ell],j\in[w]}$.

$\mathsf{KeyGen}(\mathsf{MSK}, f \in \mathcal{F})$**:** Choose one cell from each column uniformly at random; formally, choose a set $\{r_1, \ldots, r_w\} \xleftarrow{R} [\ell]^w$. Next, for each column $j \in [w]$, set $\mathsf{sk}_{r_j,j} \leftarrow \text{1-ABE.KeyGen}(\mathsf{MSK}_{r_j,j}, f)$. Output $\mathsf{sk}_f = \{r_j, \mathsf{sk}_{r_j,j}\}_{j\in[w]}$.

$\mathsf{Enc}(\mathsf{EK}, m \in \mathcal{M}, \mathsf{ind} \in \mathcal{I})$**:** Perform the following steps:

1. Perform a w-of-w secret sharing of m; formally, choose $z_1, \ldots, z_w \xleftarrow{R} \mathcal{M}$ uniformly such that $\sum_{j=1}^{w} z_j = m$. (Note that due to the finite group requirement described above, $|z_j| = |m|$ for all j.)
2. Compute the set of ciphertexts $\mathsf{ct}_{i,j} \leftarrow \text{1-ABE.Enc}(\mathsf{EK}_{i,j}, z_j, \mathsf{ind})$ for each row $i \in [\ell]$ and column $j \in [w]$,
3. Output the concatenation of $\ell \cdot w$ ciphertexts $\mathsf{ct} = \{\mathsf{ct}_{i,j}\}_{i\in[\ell],j\in[w]}$.

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$**:** Perform the following steps:

1. Parse sk_f as $\{r_j, \mathsf{sk}_{r_j,j}\}_{j\in[w]}$ and parse ct as $\{\mathsf{ct}_{i,j}\}_{i\in[\ell],j\in[w]}$.
2. For each column $j \in [w]$, let $z_j \leftarrow \text{1-ABE.Dec}(\mathsf{sk}_{r_j,j}, \mathsf{ct}_{r_j,j})$.
3. If any $z_j = \perp$, then output $\perp$. Otherwise, output $m = \sum_{j=1}^{w} z_j$.

Correctness. Suppose that a user receives a ciphertext $\mathsf{ct} = \mathsf{Enc}(\mathsf{EK}, m, \mathsf{ind})$ and she possesses a secret key $\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, f)$ for a predicate f such that $f(\mathsf{ind}) = 1$. For each column $j \in [w]$, the user possesses some secret key $\mathsf{sk}_{r_j,j}$; by the correctness of the underlying 1-ABE scheme, this key suffices to decrypt the message z_j contained in the ciphertext $\text{1-ABE.Enc}(\mathsf{EK}_{i,j}, z_j, \mathsf{ind})$. Finally, from all of the secret shares, the user may recover the original message $m = \sum_{j=1}^{w} z_j$.

As the scheme is written, repeated key queries would count as separate queries towards the bound q. In order to avoid this, the values $\{r_1, \ldots, r_w\}$ in KeyGen can be chosen pseudorandomly based on the predicate f so that the same key is issued for repeated key queries. This conversion is straightforward and we omit the details.

4.2 Setting the Parameters

The following combinatorial lemma provides a good setting of the parameters ℓ and w. We first define two probabilistic events about any set of up to q key queries made to the q-ABE scheme. Let Bad_j denote the event that there exists a row $i \in [\ell]$ such that the key query responses include two or more keys corresponding to $\mathsf{MSK}_{i,j}$. Additionally, let Bad denote the event that Bad_j occurs for all columns $j \in [w]$.

Lemma 1. *Let the* q-ABE *scheme be instantiated with* $\ell = q^2$ *and* $w = \lambda$*, and suppose at most* q KeyGen *queries are made. Then* $\Pr[\mathsf{Bad}] \leq \mathsf{negl}(\lambda)$.

Proof. Consider a single column $j \in [w]$. Note that each sk_f contains exactly one 1-ABE key $\mathsf{sk}_{r_j,j}$ for that value of j, where r_j is chosen randomly. Thus, the probability that q such values are all distinct is

$$1 \cdot \left(1 - \frac{1}{\ell}\right) \cdot \left(1 - \frac{2}{\ell}\right) \cdot \ldots \cdot \left(1 - \frac{q-1}{\ell}\right) \geq \left(1 - \frac{q-1}{\ell}\right)^q.$$

Thus, for a given column j, the event Bad_j holds with probability at most $(1 - (1 - \frac{q-1}{\ell})^q)$. The probability of Bad_j is independent for each j, so the probability that Bad_j holds for all w columns is at most $(1 - (1 - \frac{q-1}{\ell})^q)^w$. Letting $\ell = q^2$ and $w = \lambda$, we find that Bad occurs with probability at most

$$\left(1 - \left(1 - \frac{q-1}{q^2}\right)^q\right)^\lambda = \left(1 - \left(1 - \frac{1}{q} + \frac{1}{q^2}\right)^q\right)^\lambda < \left(1 - e^{-1}\right)^\lambda \leq \mathsf{negl}(\lambda),$$

where the first inequality follows from the fact that $(1 - \frac{1}{x} + \frac{1}{x^2})^x > 1/e$ for all $x > 0$.

Setting ℓ and w as indicated in Lemma 1, we arrive at the following performance characteristics for our q-ABE construction.

- MPK and MSK consist of $O(q^2\lambda)$ 1-ABE keys.
- The ciphertext size is $O(q^2\lambda)$ 1-ABE ciphertexts.
- Each decryption key has $O(\lambda)$ 1-ABE secret keys.

4.3 Security

We now prove that the q-ABE scheme defined in Sect. 4.1 is q-query secure if the underlying 1-ABE scheme is 1-query secure.

Theorem 2. *Let* 1-ABE *be any public-key (respectively, symmetric-key) ABE scheme that is* 1*-query secure. For any valid* PPT *ABE adversary* $\mathcal{A}$ *for the resulting public-key (resp., symmetric-key) scheme* q-ABE *making at most* q *key queries, there exists a valid* PPT *ABE adversary* $\mathcal{B}$ *for* 1-ABE *making at most 1 key query, with advantage* $Adv_{\text{1-ABE},\mathcal{B},1}(\lambda) \geq \frac{1}{q^2\lambda} Adv_{\text{q-ABE},\mathcal{A},q}(\lambda) - \mathsf{negl}(\lambda)$.

Proof. Let $\mathcal{A}$ be an adversary against our q-ABE construction that makes at most q key queries. We begin with the observation that the event Bad (and also all Bad_j events) depends only on the randomness tape of q-ABE.KeyGen (which chooses the random values r_j), and *not* on the values fed in as input. For the rest of this proof, we restrict KeyGen only to use randomness tapes that will not lead to the event Bad within the first q key oracle queries, so that in particular adversary $\mathcal{A}$ never causes the event Bad. Denote $\mathcal{A}$'s advantage in this modified security game as $\mathrm{Adv}'_{\mathsf{q\text{-}ABE},\mathcal{A},q}$. Since $\Pr[\mathsf{Bad}]$ is negligible by Lemma 1, our restriction causes at most negligible change to our distinguishing advantage by a standard reasoning up to failure argument:

$$\mathrm{Adv}'_{\mathsf{q\text{-}ABE},\mathcal{A},q}(\lambda) \geq \mathrm{Adv}_{\mathsf{q\text{-}ABE},\mathcal{A},q}(\lambda) - 2 \cdot \Pr[\mathsf{Bad}]. \tag{1}$$

Given some column $j^* \in [w]$, we consider a series of hybrid experiments $\mathcal{H}_0^{j^*}, \mathcal{H}_1^{j^*}, \ldots, \mathcal{H}_\ell^{j^*}$. Each experiment $\mathcal{H}_k^{j^*}$ is defined to use the same Setup and KeyGen as the modified q-ABE game, but it responds to Enc_b oracle queries $((m_0, m_1), \mathsf{ind})$ by forming the ciphertext in a special way:

Choose z_j uniformly at random for all $j \in [w], j \neq j^*$. Let $z_{j^*,0} = m_0 - \sum_{j \neq j^*} z_j$ and $z_{j^*,1} = m_1 - \sum_{j \neq j^*} z_j$.

- For $j \neq j^*$, for all i let $\mathsf{ct}_{i,j} \leftarrow \text{1-ABE.Enc}(\mathsf{EK}_{i,j}, z_j, \mathsf{ind})$.
- For $i > k$, let $\mathsf{ct}_{i,j^*} \leftarrow \text{1-ABE.Enc}(\mathsf{EK}_{i,j^*}, z_{j^*,0}, \mathsf{ind})$.
- For $i \leq k$, let $\mathsf{ct}_{i,j^*} \leftarrow \text{1-ABE.Enc}(\mathsf{EK}_{i,j^*}, z_{j^*,1}, \mathsf{ind})$.

Finally, the modified Enc_b oracle outputs $\mathsf{ct} = \{\mathsf{ct}_{i,j}\}_{i \in [\ell], j \in [w]}$.

Note that for all j^*, $\mathcal{H}_0^{j^*}$ corresponds exactly to the modified ABE security game with the encryption oracle being Enc_0, and $\mathcal{H}_\ell^{j^*}$ corresponds exactly to the modified ABE security game with the encryption oracle being Enc_1. Let $\varepsilon = \mathrm{Adv}'_{\mathsf{q\text{-}ABE},\mathcal{A},q}(\lambda)$, and let p_k denote the probability that $\mathcal{A}$ outputs 1 in experiment $\mathcal{H}_k^{j^*}$. Then $\varepsilon = |p_\ell - p_0| \leq \sum_{k=1}^{\ell} |p_k - p_{k-1}|$, so there must exist some k such that $|p_k - p_{k-1}| \geq \varepsilon/\ell$.

We now construct an adversary $\mathcal{B}$ for 1-ABE that breaks 1-query security. $\mathcal{B}$ first samples $j^* \in [w]$ uniformly at random, and chooses row $i^* \in [\ell]$ such that $|p_{i^*} - p_{i^*-1}| \geq \varepsilon/\ell$ for the chosen j^*. $\mathcal{B}$ then plays its game and interacts with $\mathcal{A}$ as follows.

Setup. $\mathcal{B}$ sets MPK_{i^*,j^*} as the public key it receives from its 1-ABE game. For all i, j such that $i \neq i^*$ or $j \neq j^*$, $\mathcal{B}$ sets $(\mathsf{MPK}_{i,j}, \mathsf{MSK}_{i,j}) \leftarrow \text{1-ABE.Setup}(1^\lambda)$.

Simulating the KeyGen oracle. When $\mathcal{A}$ makes a query to KeyGen for predicate f, $\mathcal{B}$ honestly runs q-ABE.KeyGen, with two exceptions. First, if the value r_{j^*} randomly chosen within q-ABE.KeyGen returns the value i^*, then $\mathcal{B}$ queries f to its 1-ABE KeyGen oracle and sets sk_{i^*,j^*} to be the result. Second, if the event Bad_{j^*} occurs, then $\mathcal{B}$ aborts execution of $\mathcal{A}$ and outputs a random guess in its game.

Simulating the Enc_b oracle. $\mathcal{B}$ responds to any encryption oracle query by $\mathcal{A}$ of the form $((m_0, m_1), \mathsf{ind})$ as follows. First, $\mathcal{B}$ chooses z_j uniformly at random for all $j \in [w], j \neq j^*$. Let $z_{j^*,0} = m_0 - \sum_{j \neq j^*} z_j$ and $z_{j^*,1} = m_1 - \sum_{j \neq j^*} z_j$. $\mathcal{B}$ constructs the oracle response as follows:

- For $j \neq j^*$, for all i let $\mathsf{ct}_{i,j} \leftarrow \mathsf{1\text{-}ABE.Enc}(\mathsf{EK}_{i,j}, z_j, \mathsf{ind})$.
- For $i > i^*$, let $\mathsf{ct}_{i,j^*} \leftarrow \mathsf{1\text{-}ABE.Enc}(\mathsf{EK}_{i,j^*}, z_{j^*,0}, \mathsf{ind})$.
- For $i < i^*$, let $\mathsf{ct}_{i,j^*} \leftarrow \mathsf{1\text{-}ABE.Enc}(\mathsf{EK}_{i,j^*}, z_{j^*,1}, \mathsf{ind})$.
- Query $((z_{j^*,0}, z_{j^*,1}), \mathsf{ind})$ to the Enc_b oracle of the 1-ABE game, and set ct_{i^*,j^*} to be the result.

  Output $\mathsf{ct} = \{\mathsf{ct}_{i,j}\}_{i\in[\ell],j\in[w]}$.

Guess. $\mathcal{B}$ outputs the same guess as $\mathcal{A}$.

We argue that since $\mathcal{A}$ is a valid ABE adversary, $\mathcal{B}$ is also a valid ABE adversary. Since $\mathcal{A}$ is valid, for all $((m_0, m_1), \mathsf{ind})$ queried to Enc_b and f queried to KeyGen, we have that $|m_0| = |m_1|$ and that if $f(\mathsf{ind}) = 1$, then $m_0 = m_1$. It follows that $\mathcal{B}$ is a valid ABE adversary: all shares are generated to be the same length as the secret-shared message, so $|z_{j^*,0}| = |z_{j^*,1}|$. The same f and ind are passed through to $\mathcal{B}$'s game, so if $f(\mathsf{ind}) = 1$ in $\mathcal{B}$'s queries, then $f(\mathsf{ind}) = 1$ in $\mathcal{A}$'s queries and $m_0 = m_1$, which means that the same shares are generated for the two messages, i.e., $z_{j^*,0} = z_{j^*,1}$. Furthermore, by construction, $\mathcal{B}$ queries its 1-ABE KeyGen oracle at most once.

Next, we return to the assumption from the beginning of this proof: by construction of the KeyGen oracle, the event Bad cannot occur for $\mathcal{A}$, i.e., there exists at least one column that is not bad. As a result, with probability at least $1/w$ the event Bad_{j^*} does not occur. Additionally, because the event Bad_{j^*} is independent of the specific calls made to KeyGen, it is equally likely to occur in experiments $\mathcal{H}^{j^*}_{i^*-1}$ and $\mathcal{H}^{j^*}_{i^*}$.

If the event Bad_{j^*} occurs, then $\mathcal{B}$ has no distinguishing advantage in its game by construction. Conversely, if the event Bad_{j^*} does not occur, then $\mathcal{B}$'s simulation of all oracles is faithful since $\mathcal{B}$ does not abort. Furthermore, when $b = 0$, $\mathcal{B}$ perfectly simulates $\mathcal{H}^{j^*}_{i^*-1}$, and when $b = 1$, $\mathcal{B}$ perfectly simulates $\mathcal{H}^{j^*}_{i^*}$. Putting everything together, we have

$$\mathrm{Adv}_{\mathsf{1\text{-}ABE},\mathcal{B},1}(\lambda) \geq \frac{1}{w} \cdot |p_{i^*} - p_{i^*-1}| \geq \frac{1}{\ell w}\mathrm{Adv}'_{\mathsf{q\text{-}ABE},\mathcal{A},q}(\lambda),$$

which, combined with inequality (1) and using the values of ℓ and w from Lemma 1, completes the proof.

5 Improved BC-ABE Construction

We can improve the asymptotic parameters of the above construction by performing another level of secret-sharing of each z_j. Instead of simply performing ℓ independent 1-ABE encryptions, we can reshare the z_j values once more, and then encrypt *those* shares using the 1-ABE scheme. If this new resharing were a simple linear scheme it would be equivalent to the first construction; instead, we will arrange these shares in a *multidimensional* structure.

This multidimensional secret-sharing will be created to satisfy the following two properties. First, there exist small sets of shares that are able to reconstruct.

Second, such sets of shares are rare, such that any party who only possesses the ability to obtain random shares will need to collect many shares to reconstruct.

When we use the multidimensional secret-sharing inside of our BC-ABE construction, the small sets of shares will correspond to the secret keys, yielding very short secret keys. Intuitively, security will be achieved by ensuring that the set of shares revealed when the adversary exceeds the collusion bound of the underlying 1-ABE schemes is effectively distributed randomly.

5.1 Multidimensional Secret-Sharing

In this section, we provide a multidimensional secret-sharing system. While we only use the scheme toward an improved BC-ABE construction, we codify it separately in this section because it may be of independent interest.

Definition 5 (Multidimensional secret-sharing). *Given a message y, we construct a multidimensional secret sharing scheme* $\mathsf{MultiSS}_{s,d}(y)$ *that outputs s^d shares $\sigma_{[1,1,\ldots,1]}, \ldots, \sigma_{[s,s,\ldots,s]}$ produced as follows.*

1. *Choose $s \cdot d$ "intermediate" shares $\rho_{1,1}, \ldots, \rho_{d,s}$ uniformly at random such that $\sum_{h\in[d],i\in[s]} \rho_{h,i} = y$. That is, the ρ's form a sd-of-sd secret sharing of y.*
2. *For each $\boldsymbol{v} \in [s]^d$, form the share $\sigma_{\boldsymbol{v}} = \sum_{i=1}^{d} \rho_{i,\boldsymbol{v}[i]}$.*

We can visualize the sharing in terms of a d-dimensional hypercube of side length s, where the shares $\sigma_{\boldsymbol{v}}$ are points whose coordinates are given by their subscript $\boldsymbol{v}$. Each value $\rho_{h,i}$ influences a $(d-1)$-dimensional *slice* of the hypercube—namely, it is a summand in the computation of the σ values whose h-th coordinate equals i. See Fig. 2 for a graphical representation of a three-dimensional secret-sharing scheme (i.e., $d = 3$).

We observe that a carefully-chosen set of s shares suffice to recover the original message y.

Definition 6. *Let $V = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{|V|}\}$ be a set containing vectors in $[s]^d$. We call this set* spanning *if it has the property that for each dimension $h \in [d]$, the list $(\boldsymbol{v}_1[h], \ldots, \boldsymbol{v}_{|V|}[h])$ contains all elements in $[s]$.*

If $|V| = s$, then we call this set minimally spanning. *In this case, the list $(\boldsymbol{v}_1[h], \ldots, \boldsymbol{v}_{|V|}[h])$ is a permutation of $[s]$.*

Lemma 2 (Correctness of MultiSS). *Let $\{\sigma_{\boldsymbol{v}}\}_{\boldsymbol{v}\in[s]^d} \leftarrow \mathsf{MultiSS}_{s,d}(y)$ be a multidimensional secret-sharing of y, and let V be any minimally spanning set. Then, the message y may be recovered from the s shares $\{\sigma_{\boldsymbol{v}}\}_{\boldsymbol{v}\in V}$.*

Proof. The sum $\sum_{\boldsymbol{v}\in V} \sigma_{\boldsymbol{v}}$ includes each $\rho_{h,i}$ term exactly once, so it sums to y.

Security provided by a multidimensional secret-sharing of y is captured in the following lemma.

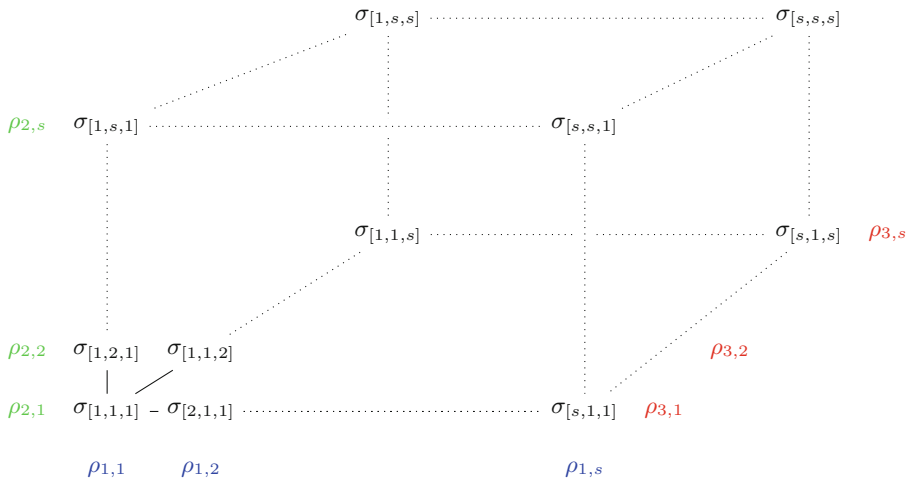

Fig. 2. Visualization of a three-dimensional secret-sharing scheme $\mathsf{MultiSS}_{s,3}(y)$. The input value y is additively secret-shared into $3s$ values $\rho_{1,1}, \ldots, \rho_{3,s}$. Each intermediate value ρ contributes to a 2-dimensional planar face of the 3-dimensional cube in which one of the dimensions is fixed to a given value (as specified by the indices to ρ). Concretely, we construct each of the s^3 shares as $\sigma_{[t,u,v]} = \rho_{1,t} + \rho_{2,u} + \rho_{3,v}$.

Lemma 3 (Security of MultiSS). *Let $\{\sigma_{\boldsymbol{v}}\}_{\boldsymbol{v}\in[s]^d} \leftarrow \mathsf{MultiSS}_{s,d}(y)$ be a multi-dimensional secret-sharing of y, and let $V^* \subseteq [s]^d$ be any set of vectors that is* not spanning. *Then, the set of shares $\{\sigma_{\boldsymbol{v}^*} : \boldsymbol{v}^* \in V^*\}$ information-theoretically reveals no information about y.*

Proof. Because $\boldsymbol{v}^*$ is not spanning, there exist a dimension $h \in [d]$ and value $i \in [s]$ such that $\boldsymbol{v}^*[h] \neq i$ for all vectors $\boldsymbol{v}^* \in V^*$. Thus, none of the shares $\{\sigma_{\boldsymbol{v}^*} : \boldsymbol{v}^* \in V^*\}$ depend on the "intermediate" share $\rho_{h,i}$, implying that $\{\sigma_{\boldsymbol{v}^*}\}$ reveals no information about y.

5.2 Construction

This construction uses similar ideas to the basic construction, with the addition of multidimensional secret-sharing. Essentially,

- The message m is additively secret-shared into $m = \sum_j z_j$, as before.
- Each of the shares z_j is multidimensionally secret-shared to form a series of s^d shares denoted by $\sigma^j_{\boldsymbol{v}}$ for $\boldsymbol{v} \in [s]^d$.
- Each $\sigma^j_{\boldsymbol{v}}$ share is encrypted using a 1-query ABE scheme in a black-box manner, producing a total of $s^d w$ resulting 1-ABE ciphertexts.

Users are given a set of keys that enable them to recover a specifically-crafted subset of the shares. If the predicate is satisfied by the index, that subset will be sufficient to reconstruct the original value at each stage of the sharing, ultimately

recovering the message. On the other hand, the multidimensional sharing step ensures that a random subset of the shares will likely need to be very large in order to recover the message. We thus gain additional collusion resistance, since the locations where collisions occur are effectively random.

Formally, let 1-ABE be a 1-query ABE scheme whose message space $\mathcal{M}$ is a finite group represented additively; we again require that $\mathcal{M}$ have the property that the set of elements of each length form a finite group. Our improved q-query secure CP-ABE scheme q-ABE* is defined below; it uses $s^d w$ independent instances of the 1-ABE scheme, where $s(\lambda)$, $d(\lambda)$, and $w(\lambda)$ are parameters that are specified later in Sect. 5.3.

$\mathsf{Setup}(1^\lambda, q)$**:** For $\boldsymbol{v} \in [s]^d$ and $j \in [w]$, let $(\mathsf{MPK}^j_{\boldsymbol{v}}, \mathsf{MSK}^j_{\boldsymbol{v}}) \leftarrow$ 1-ABE.$\mathsf{Setup}(1^\lambda)$. Output $\mathsf{MPK} = \{\mathsf{MPK}^j_{\boldsymbol{v}}\}_{\boldsymbol{v}\in[s]^d, j\in[w]}$ and $\mathsf{MSK} = \{\mathsf{MSK}^j_{\boldsymbol{v}}\}_{\boldsymbol{v}\in[s]^d, j\in[w]}$.

$\mathsf{KeyGen}(\mathsf{MSK}, f \in \mathcal{F})$**:** For each $j \in [w]$, choose a set of d permutations of $[s]$ uniformly at random. Transpose them to produce a minimally spanning set of s vectors V^j. Sample a 1-ABE key $\mathsf{sk}^j_{\boldsymbol{v}} \leftarrow$ 1-ABE.$\mathsf{KeyGen}(\mathsf{MSK}^j_{\boldsymbol{v}}, f)$ for each $j \in [w]$ and $\boldsymbol{v} \in V^j$. Finally, output $\mathsf{sk}_f = \{V^j, \{\mathsf{sk}^j_{\boldsymbol{v}}\}_{\boldsymbol{v}\in V^j}\}_{j\in[w]}$.

$\mathsf{Enc}(\mathsf{EK}, m \in \mathcal{M}, \mathsf{ind} \in \mathcal{I})$**:** Perform the following steps:

1. Perform a w-of-w additive secret-sharing of m to get shares $z_1, \ldots, z_w$ such that $\sum_{j\in[w]} z_j = m$.
2. Multidimensionally secret-share each z_j with d dimensions and s values in each dimension to create s^d shares $\{\sigma^j_{\boldsymbol{v}}\}_{\boldsymbol{v}\in[s]^d} \leftarrow \mathsf{MultiSS}_{s,d}(z_j)$.
3. For each $\boldsymbol{v} \in [s]^d$, $j \in [w]$, set $\mathsf{ct}^j_{\boldsymbol{v}} \leftarrow$ 1-ABE.$\mathsf{Enc}(\mathsf{EK}^j_{\boldsymbol{v}}, \sigma^j_{\boldsymbol{v}}, \mathsf{ind})$.
4. Output $\mathsf{ct} = \{\mathsf{ct}^j_{\boldsymbol{v}}\}_{\boldsymbol{v}\in[s]^d, j\in[w]}$.

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$**:** Perform the following steps:

1. Parse sk_f as $\{V^j, \{\mathsf{sk}^j_{\boldsymbol{v}}\}_{\boldsymbol{v}\in V^j}\}_{j\in[w]}$ and parse ct as $\{\mathsf{ct}^j_{\boldsymbol{v}}\}_{\boldsymbol{v}\in[s]^d, j\in[w]}$.
2. For each $j \in [w]$ and each $\boldsymbol{v} \in V^j$, let $\sigma^j_{\boldsymbol{v}} \leftarrow$ 1-ABE.$\mathsf{Dec}(\mathsf{sk}^j_{\boldsymbol{v}}, \mathsf{ct}^j_{\boldsymbol{v}})$.
3. Output $m = \sum_{j\in[w], \boldsymbol{v}\in V^j} \sigma^j_{\boldsymbol{v}}$.

Correctness. Suppose that a user receives a ciphertext $\mathsf{ct} = \mathsf{Enc}(\mathsf{EK}, m, \mathsf{ind})$ and she possesses a secret key $\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, f)$ for a predicate f such that $f(\mathsf{ind}) = 1$. By the correctness of the underlying 1-ABE scheme, each 1-ABE.Dec in step 2 of q-ABE*.Dec successfully returns $\sigma^j_{\boldsymbol{v}}$. For each $j \in [w]$, we may reconstruct $z_j = \sum_{\boldsymbol{v}\in V^j} \sigma_{\boldsymbol{v}}$ since KeyGen produces a minimally spanning set V^j (cf. Lemma 2), and the sum of all z_j's equals the original message m due to the w-of-w additive secret sharing.

5.3 Setting the Parameters

The combinatorial lemma in this section provides a good setting of the parameters s, d, and w. Recall that each key query yields 1-ABE keys for a minimally spanning set of vectors in each coordinate $j \in [w]$. Intuitively, we must choose s and d to be large enough that there are several minimally spanning sets, so that KeyGen rarely chooses the same vector twice. Specifically, the set of *replicated* vectors across q key queries must not be spanning.

Formally, fix some index $j \in [w]$ and consider $\mathcal{A}$'s ability to learn the j^{th} secret share $z_j = \sum_{h\in[d],i\in[s]} \rho^j_{h,i}$. The adversary $\mathcal{A}$ makes up to q queries, each of which returns s keys $\mathsf{sk}^j_{\boldsymbol{v}}$ for vectors $\boldsymbol{v}$ in a randomly-chosen minimally spanning set V^j (independent of the index queried). If $\mathcal{A}$ ever receives two keys for the same $\boldsymbol{v}$, then we no longer have any security against $\sigma_{\boldsymbol{v}}$, and therefore we assume the worst-case outcome that all of the shares $\rho^j_{h,i}$ with $\boldsymbol{v}[h] = i$ have been compromised. Let $\bar{V}^j$ denote the set of all vectors that are returned in two or more key queries.

Let Good^j denote the event that there exists some $\rho^j_{h,i}$ that remains uncompromised after $\mathcal{A}$'s queries. Observe that this is precisely the event that $\bar{V}^j$ is not spanning! In this case, the additive secret sharing protects z_j and thus m as well. Finally, let Good denote the event that there exists some $j \in [w]$ for which Good^j holds.

Lemma 4. *Let s be any constant, and instantiate the* q-ABE* *scheme with $d = \lceil 2\log_s q + 1\rceil$ and $w = \lceil \frac{\lambda}{d\cdot s}\rceil$. For any adversary $\mathcal{A}$ who makes at most q* KeyGen *queries, the event* Good *holds with overwhelming probability in λ.*

Proof. First, consider a fixed $h \in [d]$, $i \in [s]$, and $j \in [w]$. We consider $\mathcal{A}$'s ability to learn $\rho^j_{h,i}$. By construction, each of $\mathcal{A}$'s key queries yields exactly one 1-ABE key $\mathsf{sk}^j_{\boldsymbol{v}}$ where $\boldsymbol{v}$ is randomly chosen subject to the constraint that $\boldsymbol{v}[h] = i$. The probability that all of these vectors $\boldsymbol{v}$ are distinct (and thus $\rho^j_{h,i}$ is uncompromised) is therefore

$$1 \times \left(1 - \frac{1}{s^{d-1}}\right) \times \left(1 - \frac{2}{s^{d-1}}\right) \times \cdots \times \left(1 - \frac{q-1}{s^{d-1}}\right) \geq \left(1 - \frac{q-1}{s^{d-1}}\right)^q.$$

This probability holds independently for all $h \in [d]$, $i \in [s]$, and $j \in [w]$. Hence, $\Pr[\mathsf{Good}] \geq 1 - [1 - (1 - \frac{q-1}{s^{d-1}})^q]^{sdw}$.

Next, if we instantiate s, d, and w with the values provided in the lemma, we find that $1 - \Pr[\mathsf{Good}]$ is negligible:

$$\left[1 - \left(1 - \frac{q-1}{s^{d-1}}\right)^q\right]^{sdw} \leq \left[1 - \left(1 - \frac{q-1}{q^2}\right)^q\right]^{\lambda} < (1 - e^{-1})^{\lambda} = \mathsf{negl}(\lambda).$$

We list below the key and ciphertext lengths produced by our construction, when instantiated with the parameters specified in Lemma 4.

- The MPK and MSK consist of $s^d \cdot w = O(\frac{q^2\lambda}{\log q})$ 1-ABE public keys.
- A secret key consists of $s \cdot w = O(\frac{\lambda}{\log q})$ 1-ABE keys.
- A single ciphertext consists of $s^d \cdot w = O(\frac{q^2\lambda}{\log q})$ 1-ABE ciphertexts.

5.4 Security

We now prove that the q-ABE* scheme defined above is q-query secure if the underlying 1-ABE scheme is 1-query secure.

Theorem 1 (Formal). *Let* 1-ABE *be any public-key (respectively, symmetric-key) ABE scheme that is 1-query secure. For any valid* PPT *adversary* $\mathcal{A}$ *for the resulting public-key (resp., symmetric-key)* q-ABE* *construction instantiated with the parameters given in Lemma 4, there exists a valid* PPT *adversary* $\mathcal{B}$ *for* 1-ABE *making at most 1 key query, with advantage* $Adv_{\text{1-ABE},\mathcal{B},1}(\lambda) \geq \frac{1}{q^2\lambda} Adv_{\text{q-ABE}^*,\mathcal{A},q}(\lambda) - \mathsf{negl}(\lambda)$.

Proof (sketch). Here, we provide a high-level description of the reduction to the security of 1-ABE. The details mostly follow the same pattern as the proof of Theorem 2, so here we highlight the differences. Lemma 4 provides the reasoning up to failure argument analogous to that of Lemma 1.

Recall that in the proof of Theorem 2 we change a valid encryption of m_0 into a valid encryption of m_1 by changing one of the additive shares (z_j values) of the final message. Since this value is encrypted using the underlying 1-ABE scheme ℓ times, we perform this change via a sequence of hybrids. Our reduction decreases the advantage of the 1-ABE adversary by a factor of ℓ due to the selection of a hybrid step and a factor of w due to the selection of a secret share z_{j^*} to target.

In the q-ABE* construction, note that the message is effectively additively shared among sdw different values $\rho^j_{h,i}$. We can thus change an encryption of m_0 into an encryption of m_1 by changing a single one of the $\rho^j_{h,i}$ values. In this case, this value is a summand in s^{d-1} of the σ values that are encrypted using the underlying 1-ABE scheme (specifically, $\sigma^j_{\boldsymbol{v}}$ where $\boldsymbol{v}[h] = i$).

We thus require a hybrid step to change each of these encryptions to an encryption of a new value reflecting the changed $\rho^j_{h,i}$; the proof is otherwise the same. The advantage of the 1-ABE adversary decreases by a factor of s^{d-1} due to the selection of a hybrid step and a factor of sdw due to the selection of $\rho^j_{h,i}$; we omit the details. Thus, $\text{Adv}_{\text{1-ABE},\mathcal{B},1}(\lambda) \geq \frac{1}{s^d dw} \text{Adv}^*_{\text{q-ABE}^*,\mathcal{A},q}(\lambda) - \mathsf{negl}(\lambda)$, and instantiating this formula with the parameters from Lemma 4 completes the proof.

6 Instantiating 1-ABE

Thus far, we have presented two schemes for transforming any 1-ABE scheme into a q-ABE scheme. To obtain a construction of bounded-collusion ABE from CPA-secure encryption, we need to instantiate 1-ABE from CPA-secure encryption. To do so, we can use the construction of Gorbunov et al. [15] and Sahai-Seyalioglu [25] for 1-query-secure functional encryption, restricting its functionality to that of attribute-based encryption.

In this section, we briefly sketch the resulting 1-ABE scheme. We assume that it has predicates describable using n bits, that is $\mathcal{F} \subseteq \{0,1\}^n$. Note that the 1-FE from Gorbunov et al. [15] and Sahai-Seyalioglu [25] uses randomized encodings [3,20], which can be instantiated using garbled circuits. For simplicity,

we will use the language of garbled circuits in this section. Given a CPA-secure encryption scheme Σ, the 1-ABE scheme operates as follows.

$\mathsf{Setup}(1^\lambda)$**:** Generate $2n$ key pairs for the public-key encryption scheme Σ to get $(\mathsf{pk}_{i,0}, \mathsf{sk}_{i,0})$ and $(\mathsf{pk}_{i,1}, \mathsf{sk}_{i,1})$ for $i \in [n]$. Output $\mathsf{MPK} \leftarrow \{\mathsf{pk}_{i,b}\}_{i\in[n],b\in\{0,1\}}$ and $\mathsf{MSK} \leftarrow \{\mathsf{sk}_{i,b}\}_{i\in[n],b\in\{0,1\}}$

$\mathsf{KeyGen}(\mathsf{MSK}, f)$**:** Let $f[i]$ denote the i-th bit of f for $i \in [n]$. Output $\mathsf{sk}_f \leftarrow \{\mathsf{sk}_{i,f[i]}\}_{i\in[n]}$.

$\mathsf{Enc}(\mathsf{MPK}, M, \mathsf{ind})$**:** Let $U_{M,\mathsf{ind}}(f)$ be a universal circuit that takes a predicate $f \in \{0,1\}^n$ and outputs M if $f(\mathsf{ind}) = 1$ and 0 otherwise. Build a garbled circuit for $U_{M,\mathsf{ind}}$. Encrypt the two labels for each wire corresponding to the predicate f: for the i-th bit of f, encrypt the 0-label under $\mathsf{pk}_{i,0}$ and the 1-label under $\mathsf{pk}_{i,1}$. Output the garbled circuit and the encrypted wire labels.

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$**:** Use sk_f to decrypt the wire labels corresponding to f. Evaluate the garbled circuit and output the result.

As Sahai and Seyalioglu [25] show, the above scheme achieves selective security for one query. Gorbunov et al. [15] show how to modify this scheme to achieve adaptive security by using a variant of non-committing encryption [8]. This increases the number of underlying PKE components of the public parameters, keys, and the label encryptions by a factor of $O(\lambda)$ due to having to encrypt λ-bit long messages.

Thus, for a predicate description of size n and using a universal circuit U, the 1-ABE scheme has the following parameters:

- The public parameters consist of $O(n\lambda)$ PKE public keys.
- Secret keys consist of $O(n\lambda)$ PKE secret keys.
- Ciphertexts consist of $O(|U|\lambda)$ bits for the garbled gates and $O(n\lambda)$ PKE ciphertexts for the encrypted wire labels.

Putting this construction together with the parameters of our improved transformation from any 1-ABE scheme to a q-ABE scheme, we arrive at the following result that crystallizes Corollary 1.

Corollary 2. *If public-key (respectively, symmetric-key) CPA-secure encryption exists, then there exists a public-key (resp., symmetric-key) q-query secure ABE scheme for predicates that are expressible using n bits and can be evaluated by a universal circuit U with the following characteristics: public parameters (resp., MSK) consisting of $O(\frac{q^2}{\log q} n\lambda^2)$ PKE public keys (resp., secret keys), secret keys consisting of $O(\frac{n}{\log q}\lambda^2)$ PKE secret keys, and ciphertexts consisting of $O(\frac{q^2}{\log q}|U|\lambda^2)$ bits plus $O(\frac{q^2}{\log q} n\lambda^2)$ PKE ciphertexts.*

Acknowledgments. We thank the anonymous reviewers for their helpful comments.

References

1. Agrawal, S., Rosen, A.: Online-offline functional encryption for bounded collusions. IACR Cryptology ePrint Archive, 2016:361 (2016)

2. Akinyele, J.A., Lehmann, C.U., Green, M.D., Pagano, M.W., Peterson, Z.N.J., Rubin, A.D.: Self-protecting electronic medical records using attribute-based encryption. Cryptology ePrint Archive, Report 2010/565 (2010). http://eprint.iacr.org/
3. Applebaum, B., Ishai, Y., Kushilevitz, E.: Computationally private randomizing polynomials and their applications. Comput. Complex. **15**(2), 115–162 (2006)
4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (S&P 2007), Oakland, California, USA, 20–23 May 2007, pp. 321–334 (2007)
5. 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). doi:10.1007/11535218_16
6. Boneh, D., Papakonstantinou, P.A., Rackoff, C., Vahlis, Y., Waters, B.: On the impossibility of basing identity based encryption on trapdoor permutations. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, 25–28 October 2008, pp. 283–292 (2008)
7. Brakerski, Z., Segev, G.: Function-private functional encryption in the private-key setting. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 306–324. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46497-7_12
8. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, 22–24 May 1996, pp. 639–648 (1996)
9. Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., Shelat, A., Vaikuntanathan, V.: Bounded CCA2-secure encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 502–518. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76900-2_31
10. Dodis, Y., Katz, J., Xu, S., Yung, M.: Key-insulated public key cryptosystems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 65–82. Springer, Heidelberg (2002). doi:10.1007/3-540-46035-7_5
11. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994). doi:10.1007/3-540-48329-2_40
12. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40084-1_27
13. Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Symposium on Theory of Computing Conference, STOC 2013, Palo Alto, CA, USA, 1–4 June 2013, pp. 555–564 (2013)
14. Goldwasser, S., Lewko, A., Wilson, D.A.: Bounded-collusion IBE from key homomorphism. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 564–581. Springer, Heidelberg (2012). doi:10.1007/978-3-642-28914-9_32
15. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption with bounded collusions via multi-party computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 162–179. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32009-5_11
16. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Symposium on Theory of Computing Conference, STOC 2013, Palo Alto, CA, USA, 1–4 June 2013, pp. 545–554 (2013)

17. Goyal, V., Kumar, V., Lokam, S., Mahmoody, M.: On black-box reductions between predicate encryption schemes. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 440–457. Springer, Heidelberg (2012). doi:10.1007/978-3-642-28914-9_25
18. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, 30 October–3 November 2006, pp. 89–98 (2006)
19. Hamlin, A., Schear, N., Shen, E., Varia, M., Yakoubov, S., Yerukhimovich, A.: Cryptography for big data security. In: Fei, H. (ed.) Big Data: Storage, Sharing, and Security (3S), pp. 241–288. CRC Press, Taylor & Francis Group, Boca Raton (2016). (Chapter 7)
20. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: a new representation with applications to round-efficient secure computation. In: 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, Redondo Beach, California, USA, 12–14 November 2000, pp. 294–304 (2000)
21. Katz, J., Yerukhimovich, A.: On black-box constructions of predicate encryption from trapdoor permutations. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 197–213. Springer, Heidelberg (2009). doi:10.1007/978-3-642-10366-7_12
22. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_4
23. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_3
24. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, 28–31 October 2007, pp. 195–203 (2007)
25. Sahai, A., Seyalioglu, H.: Worry-free encryption: functional encryption with public keys. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, 4–8 October 2010, pp. 463–472 (2010)
26. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). doi:10.1007/11426639_27
27. Tessaro, S., Wilson, D.A.: Bounded-collusion identity-based encryption from semantically-secure public-key encryption: generic constructions with short ciphertexts. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 257–274. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54631-0_15
28. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19379-8_4