Bilinear Entropy Expansion from the Decisional Linear Assumption

Lucas Kowalczyk*
Columbia University
luke@cs.columbia.edu
and
Allison Bishop Lewko**
Columbia University
allison@cs.columbia.edu

Abstract. We develop a technique inspired by pseudorandom functions that allows us to increase the entropy available for proving the security of dual system encryption schemes under the Decisional Linear Assumption. We show an application of the tool to Attribute-Based Encryption by presenting a Key-Policy ABE scheme that is fully-secure under DLIN with short public parameters.

1 Introduction

Since its conception in [31], attribute-based encryption (ABE) has served as a demonstrably fertile ground for exploring the possible tradeoffs between expressibility, security, and efficiency in cryptographically enforced access control. In addition to the potential applications it has in its own right, the primitive of attribute-based encryption has been a catalyst for the definitions and constructions of further cryptographic primitives, such as functional encryption for general circuits. The rich structure of secret keys demanded by expressive attribute-based encryption has promoted a continuing evolution of proof techniques designed to meet the challenges inherent in balancing large and complex structures on the pinhead of simple computational hardness assumptions.

The origins of attribute-based encryption can be traced back to identity-based encryption [10, 5], where users have identities that serve as public keys and secret keys are generated on demand by a master authority. A desirable notion of security for such schemes ensures resilience against arbitrary collusions among users by allowing an attacker to demand many secret keys for individual users and attack a ciphertext encrypted to any user not represented in the set of obtained keys. Proving this kind of security requires a reduction design that can satisfy the attacker's demands without fully knowing the master secret key. This challenge is exacerbated in the (key-policy) attribute-based setting, where

* Lucas Kowalczyk is supported in part by NSF CNS 1445424 and the Office of Naval Research (ONR) contract number N00014-14-C-0113.

** Allison Lewko is supported in part by NSF CNS 1413971 and NSF CCF 1423306.

user keys correspond to access policies expressed over attributes and ciphertexts are associated with subsets of these attributes. Decryption is allowed precisely when a single user's policy is satisfied by a ciphertext's attribute set. Thus, the structure of allowable keys that the attacker can request grows more complex as the scheme is equipped to express more complex policies.

As a consequence of this, the intuitive and elegant constructions of attribute-based encryption in bilinear groups in [17, 33] were only proven secure in the selective security model: a weakened model of security that requires the attacker to declare the target of attack in advance, before seeing the public parameters of the system. This limitation of the model allows the security reduction to embed the computational challenge into its view of the public parameters of the scheme in a way that partitions the space of secret keys. Keys that do not satisfy the targeted ciphertext are able to be generated under the embedding, while keys that do satisfy the ciphertext cannot be generated. This approach does not extend well to the full security model, where this artificial limitation on the attacker is lifted.

The first fully secure ABE schemes appeared in [18], using the dual system encryption methodology [32] for designing the security reduction. In a dual system approach, there are typically multiple (computationally indistinguishable) forms of keys and ciphertexts. There are "normal" keys and ciphertexts that are employed in the real system, and then are various forms of "semi-functional" keys and ciphertexts. The core idea is to prove security via a hybrid argument, where the ciphertext is changed to semi-functional and keys are changed to semi-functional types one by one, until all the keys are of a semi-functional type incapable of decrypting the semi-functional ciphertext (it is important that they still decrypt normal ciphertexts, otherwise the hybrid transitions could be detected by the attacker who can create normal ciphertexts for itself using the public parameters). Once we reach a state where the key and ciphertexts distributions provided to the attacker are no longer bound by correct decrypt behavior, it is easier for the reduction to produce these without knowing the master secret key.

The most critical step of these dual system arguments occurs when a particular key changes from a type that can decrypt the challenge ciphertext to a type that cannot - the fact that this change is not detected by the attacker is where the reduction must use the criterion that the access policy is not satisfied. The security reductions in [18] and many subsequent works (e.g. [27, 21]) used an information-theoretic argument for this step. However, this argument requires a great deal of entropy (specifically, fresh randomness for each attribute-use in a policy). This entropy was supplied by parameters in the semi-functional space that paralleled the published parameters of the normal space. This necessitated a blowup in public parameter and ciphertext sizes, specifically a multiplicative factor of the the number of attribute-uses allowed for access policies.

In [25], it was observed that the initial steps of a typical dual system encryption hybrid argument could be re-interpreted as providing a "shadow copy" of the system parameters in the semi-functional space that does not have to be committed to when the public parameters for the normal space are provided.

This perspective suggests that one can embed a computational challenge into these semi-functional space parameters as semi-functional objects are produced. For instance, when a portion of these parameters affect a single semi-functional key that is queried after the semi-functional ciphertext, one can essentially embed the challenge in the same way as the original selective security arguments in [17]. In the reverse case, where the semi-functional key is queried before the challenge ciphertext, the embedding can be similar to a selective security proof for a ciphertext-policy ABE scheme, where keys are associated with attributes and ciphertexts are associated with access policies. In [25], state of the art selective techniques for KP-ABE and CP-ABE systems were combined into a full security proof, avoiding the blowup in parameters incurred by the information-theoretic dual system techniques.

However, even selective security for CP-ABE systems remains a rather challenging task, and the state of the art technique in [33] introduces an undesirable q-type assumption into the fully secure ABE scheme. In the CP-ABE setting, selectivity means that the attacker declares a target access policy up front. This can then be leveraged by the security reduction to design public parameters so that it can create keys precisely for sets of attributes that do not satisfy this target policy. The q-type assumption in [33] was a consequence of the need to encode a potentially large access policy into small public parameters. This leaves us still searching for an ideal KP-ABE scheme in the bilinear setting that has parameter sizes comparable to the selectively secure scheme in [17] and a full security proof from a simple assumption such as the decisional linear assumption (DLIN). A security reduction for such a scheme must seemingly break outside the mold of using either a purely information-theoretic or purely computational argument for leveraging the fact that a requested key policy cannot be satisfied by the challenge ciphertext.

Our Results To demonstrate our approach, we present a KP-ABE constructions in the composite-order bilinear setting which is proven fully secure from simple assumptions, and supports LSSS/MSP access policies (like its bilinear predecessors). Security is proven using a few specific instances of subgroup-decision assumptions and DLIN. Our scheme greatly reduces the size of the public parameters as compared to [18, 27], as the number of group elements we need to include in the public parameters grows only logarithmically rather than linearly in the bound on the number of attribute-uses in an access policy.

Our Techniques We intermix the computational and information-theoretic dual system encryption approaches, using computational steps to "boost" the entropy of a small set of (unpublished) semi-functional parameters to a level that suffices to make the prior information-theoretic argument work. Essentially, we use the fact that the semi-functional space parameters are never published to not only "delay" their definition as exploited in [25], but further to argue that they can (computationally) appear to provide more entropy than their size would information-theoretically allow. The gadget that allows us do this computational pre-processing before the running information-theoretic argument is presented as our "bilinear entropy expansion lemma."

The inspiration for the gadget construction comes from pseudorandom generators/pseudorandom functions. Naturally, if we want a small set of semi-functional generators to seemingly produce a large amount of entropy, we may want to view these parameters as the seed for a PRF, for example. Out-of-the-box PRF constructions like Naor-Reingold [26] and its DLIN-based extension [24] however are unsuitable in the bilinear setting (even though the DLIN version would remain secure) because they would require direct access to the seed for computation, and a secure bilinear construction will only provide indirect access to the seed as exponents of group elements.

To circumvent this difficulty, we use a subset-sum based construction that can be computed in a bilinear group with the seed elements in the exponents. Of course, using a naked linear structure would be detectable, but we are able to use a rather minimal amount of additional random exponents to push the linear sub-structure out of reach of detection by regular group or pairing operations.

We build our construction in two steps. First, we present a construction for a *one-use* KP-ABE system which only supports access policies where each attribute is used at most once. This scheme achieves ciphertext and key sizes which rival those of selectively secure schemes (up to constants), while significantly reducing public parameter size. Then, we apply a standard transformation to get from a one-use system to a system which allows multiple uses of attributes in policies (the number of uses allowed per attribute is constant and fixed at setup). The overhead of this transformation is drastically mitigated by our scheme's small public parameters. The effect on ciphertext and key sizes compared to previous applications of this transformation remain the same up to constants.

Further Discussion of Related Work Additional work on ABE in the bilinear setting includes various constructions of KP-ABE and CP-ABE schemes (e.g. [4, 30, 16]), schemes supporting multiple authorities (e.g. [6, 7, 29, 21]), and schemes supporting large attribute universes (e.g. [22, 28]). Some of the structure for randomization in our schemes is inspired by [22]. The large universe scheme in [28] also achieves full-security with short public parameters using conceptually different techniques. We view the main contribution of this paper to be the entropy expansion lemma, which we believe is modular and potentially useful in other settings. Our approach lends a clear understanding of the roles of information-theoretic and computational techniques in dual-system encryption proofs.

There are also recent constructions of ABE schemes in the lattice setting. The construction of [15] allows access policies to be expressed as circuits, which makes it more expressive than any known bilinear scheme. It was proven selectively secure under the standard LWE assumption. Circuit policies are also supported by the construction in [12] based on multilinear maps. This scheme is also proven selectively secure, under a particular computational hardness assumption for multilinear groups. The very recent multilinear scheme in [13] achieves full security, relying on computational hardness assumptions in multilinear groups.

The fully secure general functional encryption scheme in [34], which relies on indistinguishability obfuscation, can also be specialized to the ABE setting.

Some relationships between ABE and other cryptographic primitives have also been explored. The work of [2] derives schemes for verifiable computation from attribute-based encryption schemes, while [14] use attribute-based encryption as a tool in designing more general functional encryption and reusable garbling schemes. Dual system encryption proof techniques have also been further studied in the works of [20, 9, 34, 1], applied to achieve leakage resilience in [23, 19, 11], and applied directly to computational assumptions in [8].

2 Preliminaries

Our construction uses composite order bilinear groups. Background on these groups and the (static) subgroup decision assumptions on which our composite order construction's security is based can be found in the full version of this paper. We now give required background material on Linear Secret Sharing Schemes. The formal definition of a KP-ABE scheme, and the security definition we will use can be found in the full version.

Linear Secret Sharing Schemes Our construction uses linear secret-sharing schemes (LSSS). We use the following definition (adapted from [3]). In the context of ABE, attributes will play the role of parties and will be represented as nonempty subsets $K \subseteq [k]$ for a fixed k.

Definition 1. *(Linear Secret-Sharing Schemes (LSSS)) A secret sharing scheme Π over a set of attributes is called* linear *(over $\mathbb{Z}_p$) if the shares belonging to all attributes form a vector over $\mathbb{Z}_p$ and there exists an $\ell \times n$ matrix Λ called the share-generating matrix for Π. The matrix Λ has ℓ rows and n columns. For all $j = 1, \ldots, \ell$, the j^{th} row of Λ is labeled by an attribute K. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then Λv is the vector of ℓ shares of the secret s according to Π. The share $(\Lambda v)_j = \lambda_K$ belongs to attribute K.*

We note the *linear reconstruction* property: we suppose that Π is an LSSS. We let S denote an authorized set. Then there is a subset $S^* \subseteq S$ such that the vector $(1, 0, \ldots, 0)$ is in the span of rows of Λ indexed by S^*, and there exist constants $\{\omega_K \in \mathbb{Z}_p\}_{K \in S^*}$ such that, for any valid shares $\{\lambda_K\}$ of a secret s according to Π, we have: $\sum\limits_{K \in S^*} \omega_K \lambda_K = s$. These constants $\{\omega_K\}$ can be found in time polynomial in the size of the share-generating matrix Λ [3]. For unauthorized sets, no such S^*, $\{\omega_K\}$ exist.

For our construction, we will employ LSSS matrices over $\mathbb{Z}_N$, where N is a product of three distinct primes p, q, w. As in the definition above over the prime order $\mathbb{Z}_p$, we say a set of attributes S is authorized if is a subset $S^* \subseteq S$ such that the rows of the access matrix A labeled by elements of S have the vector

$(1, 0, \ldots, 0)$ in their span modulo N. In our security proof for our system, we will further assume that for an unauthorized set, the corresponding rows of A do not include the vector $(1, 0, \ldots, 0)$ in their span modulo q. We may assume this because if an adversary can produce an access matrix A over $\mathbb{Z}_N$ and an unauthorized set over $\mathbb{Z}_N$ that is authorized over $\mathbb{Z}_q$, then this can be used to produce a non-trivial factor of the group order N, which would violate our subgroup decision assumptions.

Transformation from One-Use to Multiple Use KP-ABE Given a KP-ABE scheme which is fully-secure when attributes are used at most once in access policies, we can obtain a KP-ABE scheme which is fully-secure when each attribute is used at most some constant number of times in access policies using a standard transformation. Essentially, multiple uses of an attribute are treated as new "attributes" in the one-use system. For example, if we want an attribute x to be able to be used up to k_x times in access policies, we will instantiate our one-use system with k_x "attributes" $x : 1, ..., x : k_x$. Each time we want to label a row of an access matrix A with x, we label it with $x : i$ for a new value of i. Each time we want to associate a subset S of attributes to a ciphertext, we instead use the set $S' = \{x : 1, ..., x : k_x \mid x \in S\}$. We can then employ the one-use KP-ABE scheme on this new larger set of "attributes" and retain its full security and functionality.

Clearly, this transformation comes at a cost. Typically, the ciphertext and public parameter size of the KP-ABE scheme resulting from the transformation now scale linearly with the number of attribute-*uses* allowed in access policies, not just the number of attributes. This presents a problem if one desires policies which have high reuse of attributes. Our one-use KP-ABE scheme mitigates the problem with public parameter size by featuring public parameters that scale only logarithmically with the number of attributes supported by the system, compared to the linear scaling of the fully secure KP-ABE schemes based on static assumptions in [18, 21]. Note that [28] also achieves full-security from static assumptions with short parameters, using conceptually different techniques.

3 KP-ABE Construction

Our single-use KP-ABE construction assumes a polynomially sized attribute universe $\mathcal{U}$ where attributes are non-empty subsets $K \subseteq [k]$ for some fixed k. The prior fully secure single-use KP-ABE scheme in [18] required a fresh group element to appear in the public parameters for each attribute in the universe. After using the generic transformation discussed in section 2, this results in the scheme requiring a fresh group element for each attribute-*use* allowed in access policies. As a concrete example, if one wanted to allow 9 attributes to be used up to 7 times each, one needed to have $9 \times 7 = 63$ group elements in the public parameters corresponding to this attribute. In our composite order scheme, to allow the same $63 = 2^6 - 1$ attribute-uses, we only need 2×6 group elements in the public parameters corresponding to the attribute. The way we accomplish this dramatic "compression" of public parameters is to note that the encryptor can produce 63 group elements from 6 by taking products of all non-empty subsets (these correspond to subset-sums in the exponent). More generally, given k group elements $g^{a_1}, \ldots, g^{a_k}$, we can produce $2^k - 1$ group elements by enumerating over all non-empty subsets $K \subseteq [k]$ and computing $g^{\sum_{j \in K} a_j}$. We name the resulting collection of elements g^{A_K}, where $A_K := \sum_{j \in K} a_j$. Our composite order scheme uses two parallel such subset constructions (causing the factor of 2).

These $2^k - 1$ group elements no longer look random - they have linear relationships in their exponents by construction. However, since we are assuming the decisional linear assumption is hard, if we choose $2^k - 1$ additional random exponents $\{t_K\}$, then the $2(2^k - 1)$ group elements formed as $\{g^{t_K}, g^{t_K A_K}\}$ are computationally indistinguishable from $2(2^k - 1)$ uniformly random group elements (which lack any hidden linear structures in their exponents). The proof of this is the core of bilinear entropy expansion lemma, though the full statement of the lemma includes some additional structure that is useful for linking into a KP-ABE construction. The dual system encryption framework allows us to apply this argument to the parameters in the semi-functional space, where we do not need to publish the values $\{g^{a_j}\}$. (Note that publishing these would make the structure of $\{g^{t_K}, g^{t_K A_K}\}$ detectable through applications of the bilinear map.)

Setup($\lambda, \mathcal{U}, k$) $\to$ PP, MSK The setup algorithm chooses a bilinear group G of order $N = pqw$ where p, q, w are primes. We let G_p, G_q, G_w represent the subgroup of order p, q, and w respectively in G. It then draws $\alpha \leftarrow \mathbb{Z}_N$ and random group element $g_p \in G_p$. For each $j \in [k]$, it chooses values $a_j, b_j \leftarrow \mathbb{Z}_N$. The public parameters are $N, g_p, e(g_p, g_p)^\alpha, \{g_p^{a_j}, g_p^{b_j} : j \in [k]\}$. The MSK is α and a generator g_w of G_w. Such a construction is equipped to create keys for access policies which include attributes $K \subseteq [k]$ where K is not empty.

KeyGen($MSK, \mathit{\Lambda}, PP$) $\to$ SK The key generation algorithm takes in the public parameters, master secret key, and LSSS access matrix $\mathit{\Lambda}$. First, the key generation algorithm generates $\{\lambda_K\}$: a linear sharing of α according to policy matrix $\mathit{\Lambda}$ (the reader is referred to section 2 for details). For each attribute K corresponding to a row in the policy matrix $\mathit{\Lambda}$, it then raises generator g_w to random exponents to create $g_w^{z_K}, g_w^{z'_K}, g_w^{z''_K} \in G_w$, chooses exponent $y_K \leftarrow \mathbb{Z}_N$ and computes $g_p^{A_K} = \prod_{j \in K} g_p^{a_j}$ and $g_p^{B_K} = \prod_{j \in K} g_p^{b_j}$. Note that here and throughout the rest of the description of our construction and its proof of security we will use the notation $A_K = \sum_{j \in K} a_j$ and $B_K = \sum_{j \in K} b_j$. It then outputs the secret key:

$$SK_{\mathit{\Lambda}} = \{g_p^{\lambda_K} g_p^{y_K A_K} g_w^{z_K}, \quad g_p^{y_K} g_w^{z'_K}, \quad g_p^{y_K B_K} g_w^{z''_K} : (\forall K \text{ labels} \in \mathit{\Lambda})\}$$

Encrypt$(M, S, PP) \rightarrow CT$ The encryption algorithm first draws $s \leftarrow \mathbb{Z}_N$. For each $K \in S$, the encryption algorithm draws $t_K \leftarrow \mathbb{Z}_N$ and computes $g_p^{A_K} = \prod_{j \in K} g_p^{a_j}$ and $g_p^{B_K} = \prod_{j \in K} g_p^{b_j}$. It then outputs the ciphertext:

$$CT = Me(g_p, g_p)^{\alpha s}, \quad \{g_p^s, \quad g_p^{sA_K} g_p^{t_K B_K}, \quad g_p^{t_K} : (\forall K \in S)\}$$

Decrypt$(CT, SK, PP) \rightarrow M$ We let S correspond to the set of attributes associated to ciphertext CT, and Λ be the policy matrix. If S satisfies Λ, the decryption algorithm computes suitable constants ω_K such that $\sum_{K \in S^*} \omega_K \lambda_K = \alpha$ (recall section 2). It then computes:

$$\begin{aligned}
&\prod_{K \in S^*} \left(e(g_p^s, \ \ g_p^{\lambda_K} g_p^{y_K A_K} g_w^{z_K}) \left(\frac{e(g_p^{y_K} g_w^{z'_K}, \ \ g_p^{sA_K} g_p^{t_K B_K})}{e(g_p^{t_K}, \ \ g_p^{y_K B_K} g_w^{z''_K})} \right)^{-1} \right)^{\omega_K} \\
&= \prod_{K \in S^*} \left(\frac{e(g_p, g_p)^{s\lambda_K} e(g_p, g_p)^{s y_K A_K}}{e(g_p, g_p)^{s y_K A_K}} \right)^{\omega_K} \\
&= \prod_{K \in S^*} \left(e(g_p, g_p)^{s\lambda_K} \right)^{\omega_K} = e(g_p, g_p)^{\sum_{K \in S^*} s\omega_K \lambda_K} = e(g_p, g_p)^{\alpha s}
\end{aligned}$$

The message can then be recovered by computing: $Me(g_p, g_p)^{\alpha s} / e(g_p, g_p)^{\alpha s} = M$. This demonstrates correctness of the scheme.

3.1 Security Proof Overview

Our security proof uses a hybrid argument over a sequence of games. We let Game$_{real}$ denote the real security game. The rest of the games use semi-functional keys and ciphertexts, which we describe below. We let g_q denote a fixed generator of the subgroup G_q, which will serve as the "semi-functional space."

Like a typical dual system encryption proof, we will begin by transitioning from a normal ciphertext to a semi-functional ciphertext with semi-functional components that mimic the structure of their normal counterparts. This kind of transition can be done with a basic subgroup decision assumption. We will then perform a hybrid over keys, gradually changing each one to a semi-functional form that does not properly decrypt the semi-functional ciphertext. To start, we can bring in semi-functional components for a particular key that mimic the structure of normal components, up to the constraint that the shared valued in the semi-functional space will be 0 (modulo q). Technically, this constraint arises because we will be taking a challenge term from a subgroup decision assumption that has an unknown exponent in the normal space and raising it to a share - so we have to make this a share of 0 and separately share the α value in the normal space so that the unknown exponent does not affect the correctness of the sharing in the normal space. At a higher level, this constraint explains why the simulator at this stage of the hybrid cannot solve the challenge problem for itself by test decrypting against a semi-functional ciphertext. Since the structure in the semi-functional space parallels the normal structure and the shared value here is zero, the semi-functional components will cancel out upon decryption.

So we can arrive at a stage where a key and ciphertext have semi-functional components structured just like the normal space, but with fresh parameters modulo q that are independent of the published parameters modulo p. This is a consequence of the Chinese Remainder Theorem, that ensures when we sample an exponent uniformly at random modulo N, its modulo p and modulo q reductions are independent and uniformly random in $\mathbb{Z}_p, \mathbb{Z}_q$ respectively. Since these implicit parameters in the semi-functional space are never published, we can use our bilinear entropy expansion lemma to argue that their subset-sum structure is hidden under the decisional linear assumption. This allows us to replace them with higher entropy parameters (lacking the subset-sum structure of the normal space), and then argue that the shared value in the semi-functional space is information-theoretically hidden (this is where we use that the access policy is not satisfied and that attributes are used at most once in the policy). This enables us to switch the semi-functional shares in the key to shares of a random value, now destroying correct decryption of a semi-functional ciphertext. We then remove some of the other (now unnecessary) semi-functional components of the key, to reclaim the entropy of those parameters to use in processing the next key in the hybrid. Finally, once we have reached a game where all keys are semi-functional with shares of a random secret modulo q, we can use Subgroup Decision Assumption 3 to create such keys without knowing the master secret and can hence complete the proof.

We now formally present our definitions of semi-functional ciphertexts and keys used in our hybrid proof:

Semi-functional Ciphertext We will use 3 types of semi-functional ciphertexts. To produce a semi-functional ciphertext for an attribute set S, one first calls the normal encryption algorithm to produce a normal ciphertext consisting of:

$$Me(g_p, g_p)^{\alpha s}, \quad \{g_p^s, \quad g_p^{sA_K} g_p^{t_K B_K}, \quad g_p^{t_K} : (\forall K \in S)\}$$

One then draws $\tilde{s} \leftarrow \mathbb{Z}_N$. For each $K \in S$, an exponent $\tilde{t_K} \leftarrow \mathbb{Z}_N$ is chosen. The remaining composition of the semifunctional ciphertext depends on the type of ciphertext desired:

Type 1 The semi-functional ciphertext of Type 1 is formed as:

$$Me(g_p, g_p)^{\alpha s}, \quad \{g_p^s g_q^{\tilde{s}}, \quad g_p^{sA_K} g_p^{t_K B_K} g_q^{\tilde{s}A_K} g_q^{\tilde{t}_K B_K}, \quad g_p^{t_K} g_q^{\tilde{t}_K} : (\forall K \in S)\}$$

(again, here $A_K = \sum_{j \in K} a_j$ and $B_K = \sum_{j \in K} b_j$)

Type 2 The semi-functional ciphertext of Type 2 is formed as:

$$Me(g_p, g_p)^{\alpha s}, \quad \{g_p^s g_q^{\tilde{s}}, \quad g_p^{sA_K} g_p^{t_K B_K} g_q^{\tilde{s} A_K} g_q^{\tilde{t}_K \tilde{b}_K}, \quad g_p^{t_K} g_q^{\tilde{t}_K} : (\forall K \in S)\}$$

for fixed $\tilde{b}_K \in \mathbb{Z}_N$ which are chosen uniformly at random and fixed if they do not already exist (in a semi-functional key, for instance).

Type 3 The semi-functional ciphertext of Type 3 is formed as:

$$Me(g_p, g_p)^{\alpha s}, \quad \{g_p^s g_q^{\tilde{s}}, \quad g_p^{sA_K} g_p^{t_K B_K} g_q^{\tilde{s} \tilde{a}_K} g_q^{\tilde{t}_K \tilde{b}_K}, \quad g_p^{t_K} g_q^{\tilde{t}_K} : (\forall K \in S)\}$$

for fixed $\tilde{a}_K, \tilde{b}_K \in \mathbb{Z}_N$ which are chosen uniformly at random and fixed if they do not already exist.

Semi-functional Keys We will use 7 types of semi-functional keys. To produce a semi-functional key for an access policy Λ, one first calls the normal key generation algorithm to produce a normal key consisting of:

$$\{g_p^{\lambda_K} g_p^{y_K A_K} g_w^{z_K}, \quad g_p^{y_K} g_w^{z'_K}, \quad g_p^{y_K B_K} g_w^{z''_K} : (\forall K \text{ labels} \in \Lambda)\}$$

The first 6 types of keys fall under 3 classes which have two variants each: a "Z" variant and an "R" variant. For Z-type keys one computes a linear sharing *of 0* under access policy Λ, creating shares $\tilde{\lambda}_K$. For R-type keys one computes a linear sharing of *a random element* u of $\mathbb{Z}_q$ which is fixed once it is created and used for all R-type keys. u is shared under access policy Λ, to create shares $\tilde{\lambda}_K$. The next steps depend on the class of the key:

Class 1 First compute $g_q^{A_K}$ and $g_q^{B_K}$ (where, again, A_K and B_K represent the subset-sums of a_j and b_j). For each K label in the honest key, one then draws $\tilde{y}_K \leftarrow \mathbb{Z}_N$ and forms the semi-functional key of type 1Z or 1R (depending on the sharing $\tilde{\lambda}_K$) as:

$$\{g_p^{\lambda_K} g_p^{y_K A_K} g_q^{\tilde{\lambda}_K} g_q^{\tilde{y}_K A_K} g_w^{z_K}, \quad g_p^{y_K} g_q^{\tilde{y}_K} g_w^{z'_K}, \quad g_p^{y_K B_K} g_q^{\tilde{y}_K B_K} g_w^{z''_K} : (\forall K \text{ labels} \in \Lambda)\}$$

Class 2 First compute $g_q^{A_K}$. *Random values* $\tilde{b}_K \in \mathbb{Z}_N$ are chosen if they do not already exist (in a semi-functional ciphertext, for instance) and fixed. For each K label in the honest key, one then draws $\tilde{y}_K \leftarrow \mathbb{Z}_N$ and forms the semi-functional key of type 2Z or 2R as:

$$\{g_p^{\lambda_K} g_p^{y_K A_K} g_q^{\tilde{\lambda}_K} g_q^{\tilde{y}_K A_K} g_w^{z_K}, \quad g_p^{y_K} g_q^{\tilde{y}_K} g_w^{z'_K}, \quad g_p^{y_K B_K} g_q^{\tilde{y}_K \tilde{b}_K} g_w^{z''_K} : (\forall K \text{ labels} \in \Lambda)\}$$

Class 3 Random values $\tilde{a}_K, \tilde{b}_K \in \mathbb{Z}_N$ are chosen if they do not already exist and fixed. For each K label in the honest key, one then draws $\tilde{y}_K \leftarrow \mathbb{Z}_N$ and forms the semi-functional key of type 3Z or 3R as:

$$\{g_p^{\lambda_K} g_p^{y_K A_K} g_q^{\tilde{\lambda}_K} g_q^{\tilde{y}_K \tilde{a}_K} g_w^{z_K}, \quad g_p^{y_K} g_q^{\tilde{y}_K} g_w^{z'_K}, \quad g_p^{y_K B_K} g_q^{\tilde{y}_K \tilde{b}_K} g_w^{z''_K} : (\forall K \text{ labels} \in \Lambda)\}$$

Note we now have defined 6 types of keys: 1Z, 1R, 2Z, 2R, 3Z, and 3R, where the letter (Z/R) describes whether the $\tilde{\lambda}_K$ share zero or a random element of $\mathbb{Z}_q$ respectively, and the number (1/2/3) describes whether the semi-functional analogues of the $g_p^{A_K}$ and $g_p^{B_K}$ in the G_q group are structured as subset-sums or as random elements of G_q (Class 1 keys have both $g_q^{A_K}$ and $g_q^{B_K}$. Class 2 keys have just $g_q^{A_K}$ structured, with a random element $g_q^{\tilde{b}_K}$. Class 3 keys have both replaced by random elements $g_q^{\tilde{a}_K}, g_q^{\tilde{b}_K}$ of G_q). There is one final type of key, type 4R, which does not contain any of these elements:
Type 4R Using shares $\tilde{\lambda}_K$ of u (which is randomly chosen from $\mathbb{Z}_p$ and fixed if it has not already been fixed), one forms the semi-functional key of type 4R as:

$$\{g_p^{\lambda_K} g_p^{y_K A_K} g_q^{\tilde{\lambda}_K} g_w^{z_K}, \quad g_p^{y_K} g_w^{z'_K}, \quad g_p^{y_K B_K} g_w^{z''_K} : (\forall K \text{ labels} \in \Lambda)\}$$

Proof Structure Our hybrid proof takes place over a series of games defined as follows: Letting Q denote the total number of key queries that the attacker makes, we define Game_{ℓ_1}, Game_{ℓ_2}, Game_{ℓ_3}, Game_{ℓ_4}, Game_{ℓ_5}, Game_{ℓ_6}, and Game_{ℓ_7} for $\ell = 1, ..., Q$. In each game, the first $\ell - 1$ keys are semi-functional of type 4R, and all keys after the ℓth request are normal. They differ in the construction of the ℓth key and the ciphertext as follows:
Game_{ℓ_1} In this game, the ℓth key is type 1Z and the ciphertext is type 1.
Game_{ℓ_2} In this game, the ℓth key is type 2Z and the ciphertext is type 2.
Game_{ℓ_3} In this game, the ℓth key is type 3Z and the ciphertext is type 3.
Game_{ℓ_4} In this game, the ℓth key is type 3R and the ciphertext is type 3.
Game_{ℓ_5} In this game, the ℓth key is type 2R and the ciphertext is type 2.
Game_{ℓ_6} In this game, the ℓth key is type 1R and the ciphertext is type 1.
Game_{ℓ_7} In this game, the ℓth key is type 4R and the ciphertext is type 1.

Note that under this definition, we have that in $\mathbf{Game}_{0_7}$, the ciphertext given to the attacker is type 1 and the keys are all normal.

The outer structure of our hybrid argument will progress as follows. First, we transition from Game_{real} to Game_{0_7}, then to Game_{1_1}, next to Game_{1_2}, next to Game_{1_3}, next to Game_{1_4}, next to Game_{1_5}, next to Game_{1_6}, next to Game_{1_7} and then to Game_{2_1} and so on. We then arrive at Game_{Q_7}, where the ciphertext is semifunctional of type 1 and *all* of the keys given to the attacker are type 4R. We then transition to one last game named Game_{final} which will complete our proof. Game_{final} uses a semi-functional ciphertext of a new type: type X:

Type X The semi-functional ciphertext of Type X is formed as:

$$MX, \quad \{g_p^s g_q^{\tilde{s}}, \quad g_p^{sA_K} g_p^{t_K B_K} g_q^{\tilde{s} A_K} g_q^{\tilde{t}_K B_K}, \quad g_p^{t_K} g_q^{\tilde{t}_K} : (\forall K \in S)\} \text{ for } X \leftarrow G_T$$

Game_{final} In this game, all keys are semi-functional of type 4R and the ciphertext is semi-functional of type X.

Note that a ciphertext of type X information-theoretically hides its message M because the message is multiplied by the uniform random X which is unused anywhere else. So, in Game_{final}, no polynomial time adversary will be able to achieve advantage in the security game, completing our proof. This hybrid argument is accomplished in the full version of this paper.

4 Bilinear Entropy Expansion Lemma

The main technical lemma used in our security argument is used to transition between hybrid games where semi-functional keys and ciphertexts have subset-sum structured $g_q^{B_K}$ components and games where they have random $g_q^{\tilde{b}_K}$ components. The relevant quantities in the following lemma are r_i where either r_i is a random exponent or is structured as a subset sum of c_i (which are analogous to the a_j, b_j in different applications of the lemma).

Definition 2. *Given G, a group of prime order q, and g a generator of that group, let $\mathcal{D}_1(m)$ be the distribution of:*

$$\begin{aligned}
&g^{\tilde{s}}, g^{\tilde{y}_1}, ..., g^{\tilde{y}_{M-1}},\\
&g^{\tilde{y}_1 r_1}, ..., g^{\tilde{y}_{M-1} r_{M-1}},\\
&g^{\tilde{y}_1 b_1}, ..., g^{\tilde{y}_{M-1} b_{M-1}},\\
&g^{\tilde{t}_1}, ..., g^{\tilde{t}_{M-1}},\\
&g^{\tilde{s} r_1 + \tilde{t}_1 b_1}, ..., g^{\tilde{s} r_{M-1} + \tilde{t}_{M-1} b_{M-1}}
\end{aligned}$$

where the $\tilde{y}_i, \tilde{t}_i, b_i, r_i, \tilde{s} \leftarrow \mathbb{Z}_q$ and $M = 2^m$.

Definition 3. *Given G, a group of prime order q, g a generator of that group, and $C = \{c_1, ..., c_m\}$ a set of m elements drawn uniformly at random from $\mathbb{Z}_q$, let $\mathcal{D}_2(m)$ be the distribution of the same elements (where the $\tilde{y}_i, \tilde{t}_i, b_i, \tilde{s} \leftarrow \mathbb{Z}_q$ and $M = 2^m$) EXCEPT that each $r_i = \sum_{j \in C_i} c_j$ where C_i denotes the ith indexed nonempty subset of C ($|C| = m$ and there are $M-1 = 2^m - 1$ nonempty subsets).*

We show that the distributions $\mathcal{D}_1(m)$ and $\mathcal{D}_2(m)$ are computationally indistinguishable if $m = O(\lg poly(\lambda))$ through an inductive proof, beginning with the base case of $m = 2$, where a distinguisher for $\mathcal{D}_1(2)$ and $\mathcal{D}_2(2)$ ($C = \{c_1, c_2\}$) can be used to achieve the same advantage in the 2-Linear Problem.

Lemma 1. *If there exists a polynomial-time algorithm able to achieve advantage $2^2\delta$ in distinguishing between the distributions $\mathcal{D}_1(2)$ and $\mathcal{D}_2(2)$, then there exists a polynomial-time algorithm able to achieve advantage δ in the 2-Linear Problem.*

Proof. If there exists a polynomial time algorithm $\mathcal{A}$ which distinguishes between $\mathcal{D}_1(2)$ and $\mathcal{D}_2(2)$ with advantage $2^2\delta$, we can construct a distinguisher for the 2-Linear problem: $\mathcal{B}$. $\mathcal{B}$, upon receiving $g, g^{y_1}, g^{y_2}, g^{y_1 c_1}, g^{y_2 c_2}, g^{c_1+c_2+r}$, draws uniform random $\tilde{s}, b_3, \tilde{y}_3, \tilde{t}_1, \tilde{t}_2, \tilde{t}_3, \gamma_1, \gamma_2 \leftarrow \mathbb{Z}_q$, then creates the set:

$$\begin{aligned}
&g^{\tilde{s}}, g^{y_1}, \quad g^{y_2}, \quad g^{\tilde{y}_3},\\
&g^{y_1 c_1}, \;\; g^{y_2 c_2}, \;\; (g^{c_1+c_2+r})^{\tilde{y}_3},\\
&(g^{y_1 c_1})^{-\frac{\tilde{s}}{\tilde{t}_1}} (g^{y_1})^{\gamma_1}, \;\; (g^{y_2 r_c})^{-\frac{\tilde{s}}{\tilde{t}_2}} (g^{y_2})^{\gamma_2}, \;\; g^{\tilde{y}_3 b_3},\\
&g^{\tilde{t}_1}, \quad g^{\tilde{t}_2}, \quad g^{\tilde{t}_3},\\
&g^{\tilde{t}_1 \gamma_1}, \;\; g^{\tilde{t}_2 \gamma_2}, \;\; (g^{c_1+c_2+r})^{\tilde{s}} g^{\tilde{t}_3 b_3}
\end{aligned}$$

then runs $\mathcal{A}$ on this input and returns the output of $\mathcal{A}$.

Notice that if $r = 0$, this distribution is exactly $\mathcal{D}_2(2)$ (with $C = \{c_1, c_2\}$, $\tilde{y}_1 = y_1, \tilde{y}_2 = y_2, b_1 = -\frac{c_1 s}{\tilde{t}_1} + \gamma_1$, and $b_2 = -\frac{c_2 s}{\tilde{t}_2} + \gamma_2$). If r is instead random, this distribution is exactly $\mathcal{D}_1(2)$. Therefore, $\mathcal{B}$ will achieve the same advantage $2^2\delta$ as $\mathcal{A}$ (which is greater than δ) in deciding the 2-Linear problem.

Lemma 2. *For all integers $m \geq 2$, if there exists a polynomial-time algorithm able to achieve an advantage of $2^{m+1}\delta$ deciding between distributions $\mathcal{D}_1(m+1)$ and $\mathcal{D}_2(m+1)$, then either there exists a polynomial-time algorithm able to achieve an advantage of $2^m\delta$ in deciding between distributions $\mathcal{D}_1(m)$ and $\mathcal{D}_2(m)$ or there exists a polynomial time algorithm able to achieve an advantage of δ in the 2-Linear Problem.*

Proof. If there exists a polynomial time algorithm $\mathcal{A}$ which distinguishes between $\mathcal{D}_1(m+1)$ and $\mathcal{D}_2(m+1)$ with non-negligible advantage $2^{m+1}\delta$, we construct $\mathcal{B}$: a distinguisher for $\mathcal{D}_1(m)$ and $\mathcal{D}_2(m)$.

$\mathcal{B}$, upon receiving:
$g^{\tilde{s}}, g^{\tilde{y}_1}, ..., g^{\tilde{y}_{M-1}}, g^{\tilde{y}_1 r_1}, ..., g^{\tilde{y}_{M-1} r_{M-1}}, g^{\tilde{y}_1 b_1}, ..., g^{\tilde{y}_{M-1} b_{M-1}}, g^{\tilde{t}_1}, ..., g^{\tilde{t}_{M-1}}, g^{\tilde{s} r_1 + \tilde{t}_1 b_1}, ..., g^{\tilde{s} r_{M-1} + \tilde{t}_{M-1} b_{M-1}}$
where $M = 2^m$, first draws:
$y_1^*, ..., y_{M-1}^*, \sigma_1, ..., \sigma_{M-1}, \gamma_1, ..., \gamma_{M-1}, \tilde{y}_M, \tilde{t}_M, b_M, c_{m+1} \leftarrow \mathbb{Z}_q$, and constructs:

$$g^{\tilde{s}}, g^{\tilde{y}_1}, ..., g^{\tilde{y}_{M-1}}, g^{\tilde{y}_M},$$

$$(g^{\tilde{y}_1})^{y_1^*}, ..., (g^{\tilde{y}_{M-1}})^{y_{M-1}^*},$$

$$g^{\tilde{y}_1 r_1}, ..., g^{\tilde{y}_{M-1} r_{M-1}}, g^{\tilde{y}_M c_{m+1}},$$

$$(g^{\tilde{y}_1})^{y_1^* c_{m+1}} (g^{\tilde{y}_1 r_1})^{y_1^*}, ..., (g^{\tilde{y}_{M-1}})^{y_{M-1}^* c_{m+1}} (g^{\tilde{y}_{M-1} r_{M-1}})^{y_{M-1}^*},$$

$$g^{\tilde{y}_1 b_1}, ..., g^{\tilde{y}_{M-1} b_{M-1}}, g^{\tilde{y}_M b_M},$$

$$(g^{\tilde{y}_1 b_1})^{y_1^*} (g^{\tilde{y}_1})^{\sigma_1 y_1^*}, ..., (g^{\tilde{y}_{M-1} b_{M-1}})^{y_{M-1}^*} (g^{\tilde{y}_{M-1}})^{\sigma_{M-1} y_{M-1}^*},$$

$$g^{\tilde{t}_1}, ..., g^{\tilde{t}_{M-1}}, g^{\tilde{t}_M},$$

$$g^{\tilde{t}_1} (g^{\tilde{y}_1})^{\gamma_1}, ..., g^{\tilde{t}_{M-1}} (g^{\tilde{y}_{M-1}})^{\gamma_{M-1}},$$

$$g^{\tilde{s} r_1 + \tilde{t}_1 b_1}, ..., g^{\tilde{s} r_{M-1} + \tilde{t}_{M-1} b_{M-1}}, (g^{\tilde{s}})^{c_{m+1}} g^{\tilde{t}_M b_M},$$

$$(g^{\tilde{s}})^{c_{m+1}} g^{\tilde{s} r_1 + \tilde{t}_1 b_1} (g^{\tilde{t}_1})^{\sigma_1} (g^{b_1 \tilde{y}_1})^{\gamma_1} (g^{\tilde{y}_1})^{\sigma_1 \gamma_1}, ..., (g^{\tilde{s}})^{c_{m+1}} g^{\tilde{s} r_{M-1} + \tilde{t}_{M-1} b_{M-1}} (g^{\tilde{t}_{M-1}})^{\sigma_{M-1}} (g^{b_{M-1} \tilde{y}_{M-1}})^{\gamma_{M-1}} (g^{\tilde{y}_{M-1}})^{\sigma_{M-1} \gamma_{M-1}}$$

which is equal to:

$$\begin{aligned}
&g^{\tilde{s}}, g^{\tilde{y}_1}, ..., g^{\tilde{y}_{M-1}}, g^{\tilde{y}_M},\\
&g^{\tilde{y}_1 y_1^*}, ..., g^{\tilde{y}_{M-1} y_{M-1}^*},\\
&g^{\tilde{y}_1 r_1}, ..., g^{\tilde{y}_{M-1} r_{M-1}}, g^{\tilde{y}_M c_{m+1}},\\
&g^{\tilde{y}_1 y_1^*(r_1+c_{m+1})}, ..., g^{\tilde{y}_{M-1} y_{M-1}^*(r_{M-1}+c_{m+1})},\\
&g^{\tilde{y}_1 b_1}, ..., g^{\tilde{y}_{M-1} b_{M-1}}, g^{\tilde{y}_M b_M},\\
&g^{\tilde{y}_1 y_1^*(b_1+\sigma_1)}, ..., g^{\tilde{y}_{M-1} y_{M-1}^*(b_{M-1}+\sigma_{M-1})},\\
&g^{\tilde{t}_1}, ..., g^{\tilde{t}_{M-1}}, g^{\tilde{t}_M},\\
&g^{\tilde{t}_1+\tilde{y}_1\gamma_1}, ..., g^{\tilde{t}_{M-1}+\tilde{y}_{M-1}\gamma_{M-1}},\\
&g^{\tilde{s}r_1+\tilde{t}_1 b_1}, ..., g^{\tilde{s}r_{M-1}+\tilde{t}_{M-1} b_{M-1}}, g^{\tilde{s}c_{m+1}+\tilde{t}_M b_M},\\
&g^{\tilde{s}(r_1+c_{m+1})+(\tilde{t}_1+\tilde{y}_1\gamma_1)(b_1+\sigma_1)}, ..., g^{\tilde{s}(r_{M-1}+c_{m+1})+(\tilde{t}_{M-1}+\tilde{y}_{M-1}\gamma_{M-1})(b_{M-1}+\sigma_{M-1})}
\end{aligned}$$

Notice that if $\mathcal{B}$'s input is $\mathcal{D}_2(m)$, then the distribution of sets constructed by $\mathcal{B}$ is exactly $\mathcal{D}_2(m+1)$, where a new c_{m+1} element is drawn and added to form the subsets of the new augmented set C, $\tilde{y}_{M+i} = \tilde{y}_i y_i^*$, $b_{M+i} = b_i + \sigma_i$, and $\tilde{t}_{M+i} = \tilde{t}_i + \tilde{y}_i\gamma_i$ for $i = 1, ..., M-1$ which are all uniformly distributed at random. However, if $\mathcal{B}$'s input is $\mathcal{D}_1(m)$, then the distribution of sets constructed by $\mathcal{B}$ is not exactly $\mathcal{D}_1(m+1)$.

Definition 4. *Let $\mathcal{D}_1'(m+1)$ be the distribution of sets created by $\mathcal{B}$ given input sets from $\mathcal{D}_1(m)$.*

We have therefore only proved that if an algorithm is able to achieve advantage in distinguishing $\mathcal{D}_1'(m+1)$ and $\mathcal{D}_2(m+1)$, then it can be used to achieve that same advantage in deciding between $\mathcal{D}_1(m)$ and $\mathcal{D}_2(m)$. Fortunately, we can transition between $\mathcal{D}_1'(m+1)$ and $\mathcal{D}_1(m+1)$ using a hybrid proof. First we define $M = 2^m$ hybrid distributions indexed by (j):

Definition 5. *Let $\mathcal{D}_1'^{(j)}(m+1)$ be the distribution of:*

$$\begin{aligned}
&g^{\tilde{s}}, g^{\tilde{y}_1}, ..., g^{\tilde{y}_{M-1}}, g^{\tilde{y}_M},\\
&g^{\tilde{y}_{M+1}}, ..., g^{\tilde{y}_{2M-1}},\\
&g^{\tilde{y}_1 r_1}, ..., g^{\tilde{y}_{M-1} r_{M-1}}, g^{\tilde{y}_M c_{m+1}},\\
&g^{\tilde{y}_{M+1}(r_1+c_{m+1})}, ..., g^{\tilde{y}_{M+j}(r_j+c_{m+1})}, g^{\tilde{y}_{M+j+1} r_{M+j+1}}, ..., g^{\tilde{y}_{2M-1} r_{2M-1}},\\
&g^{\tilde{y}_1 b_1}, ..., g^{\tilde{y}_{M-1} b_{M-1}}, g^{\tilde{y}_M b_M},\\
&g^{\tilde{y}_{M+1} b_{M+1}}, ..., g^{\tilde{y}_{2M-1} b_{2M-1}},\\
&g^{\tilde{t}_1}, ..., g^{\tilde{t}_{M-1}}, g^{\tilde{t}_M},\\
&g^{\tilde{t}_{M+1}}, ..., g^{\tilde{t}_{2M-1}},\\
&g^{\tilde{s}r_1+\tilde{t}_1 b_1}, ..., g^{\tilde{s}r_{M-1}+\tilde{t}_{M-1} b_{M-1}}, g^{\tilde{s}c_{m+1}+\tilde{t}_M b_M},\\
&g^{\tilde{s}(r_1+c_{m+1})+\tilde{t}_{M+1} b_{M+1}}, ..., g^{\tilde{s}(r_j+c_{m+1})+\tilde{t}_{M+j} b_{M+j}}, g^{\tilde{s}r_{M+j+1}+\tilde{t}_{M+j+1} b_{M+j+1}}, ..., g^{\tilde{s}r_{2M-1}+\tilde{t}_{2M-1} b_{2M-1}}
\end{aligned}$$

for $j = 0, ..., M-1$ *where the* r_{M+j+i} *are distributed uniformly at random in* $\mathbb{Z}_p$ *for* $i = 1, ...M-j-1$

Notice that $\mathcal{D}_1'^{(0)}(m+1) = \mathcal{D}_1(m+1)$ and $\mathcal{D}_1'^{(M-1)}(m+1) = \mathcal{D}_1'(m+1)$. So, if some adversary $\mathcal{A}$ could distinguish between $\mathcal{D}_1(m+1)$ and $\mathcal{D}_1'(m+1)$ with non-negligible advantage $2^m\delta$, then by the triangle inequality, there must exists some j such that: $\left|\Pr[\mathcal{A} = 1|\mathcal{D}_1'^{(j+1)}(m+1)] - \Pr[\mathcal{A} = 1|\mathcal{D}_1'^{(j)}(m+1)]\right| \geq \frac{2^m\delta}{M} = \delta$. Such an $\mathcal{A}$ can be used to construct a distinguisher for the 2-Linear Problem: $\mathcal{B}$ that achieves advantage δ:

$\mathcal{B}$, upon receiving $g, g^{y_1}, g^{y_2}, g^{y_1c_1}, g^{y_2c_2}, g^{c_1+c_2+r}$, relabels the elements as: $g, g^{y_1}, g^{y_2}, g^{y_1r^*}, g^{y_2c_{m+1}}, g^x$ (defining $y_1 = y_1, y_2 = y_2, r^* = c_1, c_{m+1} = c_2$, and $x = r^* + c_{m+1} + r$). $\mathcal{B}$ then draws $\tilde{s}, \gamma_{j+1}, \tilde{y}_1, ..., \tilde{y}_j, \tilde{y}_{j+2}, ..., \tilde{y}_{M-1}, y^*_M, ..., y^*_{M+j}, \tilde{y}_{M+j+1}, ..., \tilde{y}_{2M-1}$, $\tilde{t}_1, ..., \tilde{t}_{2M-1}, \gamma_M, ..., \gamma_{M+j}, b_{j+2}, ..., b_{2M-1}, r_1, ..., r_j, r_{j+2}, ..., r_{2M-1}$ uniformly at random from $\mathbb{Z}_q$ and constructs:

$$g^{\tilde{s}}, g^{\tilde{y}_1}, ..., g^{\tilde{y}_j}, g^{y_1}, g^{\tilde{y}_{j+2}}, ..., g^{\tilde{y}_{M-1}}, (g^{y_2})^{y^*_M},$$
$$(g^{y_2})^{y^*_{M+1}}, ..., (g^{y_2})^{y^*_{M+j}}, g^{\tilde{y}_{M+j+1}}, ..., g^{\tilde{y}_{2M-1}},$$
$$g^{\tilde{y}_1r_1}, ..., g^{\tilde{y}_jr_j}, g^{y_1r^*}, g^{\tilde{y}_{j+2}r_{j+2}}, ..., g^{\tilde{y}_{M-1}r_{M-1}}, (g^{y_2c_{m+1}})^{y^*_M},$$
$$((g^{y_2})^{y^*_{M+1}})^{r_1}(g^{y_2c_{m+1}})^{y^*_{M+1}}, ..., ((g^{y_2})^{y^*_{M+j}})^{r_j}(g^{y_2c_{m+1}})^{y^*_{M+j}}, (g^x)^{\tilde{y}_{M+j+1}}, g^{\tilde{y}_{M+j+2}r_{M+j+2}}, ..., g^{\tilde{y}_{2M-1}r_{2M-1}},$$
$$g^{\tilde{y}_1b_1}, ..., g^{\tilde{y}_jb_j}, (g^{y_1r^*})^{-\frac{\tilde{s}}{\tilde{t}_{j+1}}}(g^{y_1})^{\gamma_{j+1}}, g^{\tilde{y}_{j+2}b_{j+2}}, ..., g^{\tilde{y}_{M-1}b_{M-1}}, (g^{y_2})^{y^*_{M-1}b_{M-1}}, (g^{y_2})^{y^*_M\gamma_M}(g^{y_2c_{m+1}})^{-y^*_M\frac{\tilde{s}}{\tilde{t}_M}}$$
$$(g^{y_2})^{-y^*_{M+1}(\frac{\tilde{s}r_1}{\tilde{t}_{M+1}}-\gamma_{M+1})}(g^{y_2c_{m+1}})^{-y^*_{M+1}\frac{\tilde{s}}{\tilde{t}_{M+1}}}, ..., (g^{y_2})^{-y^*_{M+j}(\frac{\tilde{s}r_j}{\tilde{t}_{M+j}}-\gamma_{M+j})}(g^{y_2c_{m+1}})^{-y^*_{M+j}\frac{\tilde{s}}{\tilde{t}_{M+j}}},$$
$$g^{\tilde{y}_{M+j+1}b_{M+j+1}}, ..., g^{\tilde{y}_{2M-1}b_{2M-1}},$$
$$g^{\tilde{t}_1}, ..., g^{\tilde{t}_M},$$
$$g^{\tilde{t}_{M+1}}, ..., g^{\tilde{t}_{2M-1}},$$
$$g^{\tilde{s}r_1+\tilde{t}_1b_1}, ..., g^{\tilde{s}r_j+\tilde{t}_jb_j}, g^{\tilde{t}_{j+1}\gamma_{j+1}}, g^{\tilde{s}r_{j+1}+\tilde{t}_{j+1}b_{j+1}}, ..., g^{\tilde{s}r_{M-1}+\tilde{t}_{M-1}b_{M-1}}, g^{\tilde{t}_M\gamma_M}$$
$$g^{\tilde{t}_{M+1}\gamma_{M+1}}, ..., g^{\tilde{t}_{M+j}\gamma_{M+j}}, (g^x)^{\tilde{s}}g^{\tilde{t}_{M+j+1}b_{M+j+1}}, g^{\tilde{s}r_{M+j+2}+\tilde{t}_{M+j+2}b_{M+j+2}}, ..., g^{\tilde{s}r_{2M-1}+\tilde{t}_{2M-1}b_{2M-1}}$$

where $\tilde{y}_{j+1} = y_1$, $r_{j+1} = r^*, b_{j+1} = -\frac{\tilde{s}r^*}{\tilde{t}_{j+1}} + \gamma_{j+1}$, $b_M = -\frac{\tilde{s}c_{m+1}}{\tilde{t}_M} + \gamma_M$, and the $b_{M+i} = -\frac{\tilde{s}(r_i+c_{m+1})}{\tilde{t}_{M+i}} + \gamma_{M+i}$ for $i = 1, ..., j$ and $\tilde{y}_{M+i} = y_2\tilde{y}^*_{M+i}$ for $i = 0, ..., j$ are all distributed uniformly at random in $\mathbb{Z}_p$.

$\mathcal{B}$ then runs $\mathcal{A}$ on this input and outputs the same.

Note that if $x = r^* + c_{m+1} + 0$, then $\mathcal{B}$ has sampled an instance of $\mathcal{D}_1'^{(j+1)}(m+1)$. Otherwise, if $x = r^* + c_{m+1} + r$ for a uniform random r it has sampled an instance of $\mathcal{D}_1'^{(j)}(m+1)$. So, $\mathcal{B}$ will enjoy the same advantage δ of $\mathcal{A}$ but in deciding the 2-Linear Problem.

We assumed there is a polynomial time algorithm $\mathcal{A}$ which distinguishes between $\mathcal{D}_1(m+1)$ and $\mathcal{D}_2(m+1)$ with advantage $2^{m+1}\delta$. By the triangle inequality, then $\mathcal{A}$ must be able to be used to either achieve advantage $2^m\delta$ in distinguishing between instances of $\mathcal{D}_1(m+1)$ and $\mathcal{D}'_1(m+1)$ or achieve advantage $2^m\delta$ in distinguishing between instances of between $\mathcal{D}'_1(m+1)$ and $\mathcal{D}_2(m+1)$.

In the first case, if $\mathcal{A}$ can be used to achieve advantage $2^m\delta$ in distinguishing between instances of $\mathcal{D}_1(m+1)$ and $\mathcal{D}'_1(m+1)$, then we showed in the first proof how such an algorithm could be used to distinguish between $\mathcal{D}_1(m)$ and $\mathcal{D}_2(m)$ with the same advantage ($2^m\delta$).

In the second case, if $\mathcal{A}$ can be used to achieve advantage $2^m\delta$ in distinguishing between instances of $\mathcal{D}'_1(m+1)$ and $\mathcal{D}_2(m+1)$, then we showed in the second proof how such an algorithm could be used to break the 2-Linear problem with advantage $\frac{2^m\delta}{M} = \delta$.

Therefore, if there is a polynomial time algorithm $\mathcal{A}$ which distinguishes between $\mathcal{D}_1(m+1)$ and $\mathcal{D}_2(m+1)$ with advantage $2^{m+1}\delta$, then either there exists a polynomial-time algorithm able to achieve an advantage of $2^m\delta$ in deciding between distributions $\mathcal{D}_1(m)$ and $\mathcal{D}_2(m)$ or there exists a polynomial time algorithm able to achieve an advantage of δ in the 2-Linear Problem.

Lemma 3. *The distributions $\mathcal{D}_1(k)$ and $\mathcal{D}_2(k)$ are computationally indistinguishable under the 2-Linear computational hardness assumption if $k = O(\lg poly(\lambda))$.*

Proof. We have shown that for all integers $m \geq 2$, if there exists a polynomial-time algorithm able to achieve an advantage of $2^{m+1}\delta$ deciding between distributions $\mathcal{D}_1(m+1)$ and $\mathcal{D}_2(m+1)$, then either there exists a polynomial-time algorithm able to achieve an advantage of $2^m\delta$ in deciding between distributions $\mathcal{D}_1(m)$ and $\mathcal{D}_2(m)$ or there exists a polynomial time algorithm able to achieve an advantage of δ in the 2-Linear Problem. We have also shown that if there exists a polynomial-time algorithm able to achieve advantage $2^2\delta$ in distinguishing between the distributions $\mathcal{D}_1(2)$ and $\mathcal{D}_2(2)$, then there exists a polynomial-time algorithm able to achieve advantage δ in the 2-Linear Problem. By induction, it follows that for all m, if an algorithm is able to achieve an advantage of $2^m\delta$ in distinguishing between distributions $\mathcal{D}_1(m)$ and $\mathcal{D}_2(m)$, then that algorithm can be used to achieve advantage δ in the 2-Linear problem.

If $k = O(\lg \mathrm{poly}(\lambda))$, then any algorithm $\mathcal{A}$ able to achieve non-negligible advantage δ in distinguishing between $\mathcal{D}_1(k)$ and $\mathcal{D}_2(k)$ can be used to achieve non-negligible advantage $\Omega(\frac{\delta}{\mathrm{poly}(\lambda)})$ in the 2-Linear problem. This violates our 2-Linear Assumption, so no such algorithm $\mathcal{A}$ can exist.

5 Concluding Remarks

We have presented a composite order KP-ABE scheme proven fully secure under the DLIN assumption and additional subgroup decision type assumptions. The scheme allows a bound of $2^k - 1$ attribute-uses in an access policy, where the number of group elements required in the public parameters per attribute-use grows polynomially with k. An interesting question for future work is whether the ciphertext sizes can be significantly reduced (our scheme has ciphertexts still growing linearly in size with $2^k - 1$). We have chosen to demonstrate our techniques on a KP-ABE scheme, though we note that they are equally applicable to the CP-ABE setting. The core of CP-ABE schemes often mirror the structure of KP-ABE schemes, and would benefit similarly from the reduced public parameter size our lemma enables. Finally, our bilinear entropy expansion lemma is not restricted to the ABE setting, and we suspect it may have applications to other cryptographic primitives. Primitive structure can be built around the lemma's core components of $\{g^{t_K}, g^{t_K A_K}\}$, which can be plugged in to replace a need for independent random group elements. Our composite order KP-ABE scheme demonstrates this usage.

References

1. N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In *EUROCRYPT*, pages 557–577, 2014.
2. M. Raykova B. Parno and V. Vaikuntanathan. How to delegate and verify in public: Verifiable computation from attribute-based encryption. In *TCC*, pages 422–439, 2012.
3. A. Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
4. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 321–334.
5. D. Boneh and M. Franklin. Identity based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.
6. M. Chase. Multi-authority attribute based encryption. In *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, pages 515–534, 2007.
7. M. Chase and S. S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *Proceedings of the 2009 ACM Conference on Computer and Communications Security*, pages 121–130, 2009.
8. M. Chase and S. Meiklejohn. Déjà Q: using dual systems to revisit q-type assumptions. In *EUROCRYPT*, pages 622–639, 2014.
9. J. Chen and H. Wee. Fully, (almost) tightly secure IBE and dual system groups. In *CRYPTO*, pages 435–460, 2013.
10. C. Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 26–28, 2001.
11. Y. Dodis, A. B. Lewko, B. Waters, and D. Wichs. Storing secrets on continually leaky devices. In *FOCS*, pages 688–697, 2011.
12. S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO*, pages 479–499, 2013.
13. S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure attribute based encryption from multilinear maps. *IACR Cryptology ePrint Archive*, 2014:622, 2014.
14. S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. How to run turing machines on encrypted data. In *Advances in Cryptology -*

CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II, pages 536–553, 2013.

15. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554, 2013.
16. V. Goyal, A. Jain, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute-based encryption. In *ICALP*, 2008.
17. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute based encryption for fine-grained access control of encrypted data. In *ACM conference on Computer and Communications Security*, pages 89–98, 2006.
18. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pages 62–91, 2010.
19. A. Lewko, Y. Rouselakis, and B. Waters. Achieving leakage resilience through dual system encryption. In *TCC*, pages 70–88, 2011.
20. A. Lewko and B. Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In *TCC*, pages 455–479, 2010.
21. A. Lewko and B. Waters. Decentralizing attribute-based encryption. In *EUROCRYPT*, pages 568–588, 2011.
22. A. Lewko and B. Waters. Unbounded hibe and attribute-based encryption. In *EUROCRYPT*, pages 547–567, 2011.
23. A. B. Lewko, M. Lewko, and B. Waters. How to leak on key updates. In *STOC*, pages 725–734, 2011.
24. A. B. Lewko and B. Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In *Proceedings of the 2009 ACM Conference on Computer and Communications Security*, pages 112–120, 2009.
25. A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, pages 180–198, 2012.
26. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *FOCS*, pages 458–467, 1997.
27. T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.
28. T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. In *ASIACRYPT*, pages 349–366, 2012.
29. T. Okamoto and K. Takashima. Decentralized attribute-based signatures. In *PKC*, pages 125–142, 2013.
30. R. Ostrovksy, A. Sahai, and B. Waters. Attribute based encryption with non-monotonic access structures. In *ACM conference on Computer and Communications Security*, pages 195–203, 2007.
31. A. Sahai and B. Waters. Fuzzy identity based encryption. In *EUROCRYPT*, pages 457–473, 2005.
32. B. Waters. Dual system encryption: realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO*, pages 619–636, 2009.
33. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *PKC*, pages 53–70, 2011.
34. H. Wee. Dual system encryption via predicate encodings. In *TCC*, pages 616–637, 2014.