Mitigating the One-Use Restriction in Attribute-Based Encryption

Lucas Kowalczyk, Jiahui Liu, Tal Malkin, and Kailash Meiyappan

Columbia University
luke@cs.columbia.edu
jl4161@columbia.edu
tal@cs.columbia.edu
kkm2142@columbia.edu

Abstract. We present a key-policy attribute-based encryption scheme that is adaptively secure under a static assumption and is not directly affected by an attribute "one-use restriction." Our construction improves upon the only other such scheme (Takashima '17) by mitigating its downside of a ciphertext size that is dependent on the maximum size of any supported attribute set.

1 Introduction

Attribute-based encryption (ABE) is a type of public key encryption which allows for fine-grained access control to encrypted data. In Key-Policy ABE, ciphertexts are associated with attributes, and secret-keys are associated with Boolean access policies that take in a set of attributes and return True if the key is capable of decrypting ciphertexts associated with that set and return False otherwise. Security guarantees that (potentially colluding) users without an authorized key should not be able to learn anything about an encrypted message. (A dual variant called Ciphertext-Policy ABE swaps the roles of attributes and access policies to be associated with the secret keys and ciphertexts respectively).

One way to make security proofs for ABE more attainable is to consider restricted notions of security. For KP-ABE, the notion of *selective security* requires the adversary to commit to a target set of attributes for the challenge ciphertext that will be attacked at the start of the security game. The earliest constructions of ABE using bilinear groups were proven secure in this model [17,34].The notion of *semi-adaptive security* [18] requires the adversary to commit to a target set of attributes, but allows the adversary to see the public parameters first. These notions are obviously not realistic attack scenarios, so a KP-ABE scheme would ideally satisfy the notion of *adaptive security* (full security), where the challenge attribute set can be chosen adaptively (in response to public parameters and any amount of secret keys received). The first construction of ABE achieving adaptive security appeared in [21], employing the dual system encryption methodology [33] in its security reduction.

Another way to make proving security of ABE schemes easier is to reduce security to parameterized assumptions like *q-type assumptions*, where the size of

the elements included in the assumption's challenge grows with some property of the adversary. q-type assumptions were used in the ABE constructions of [34,24] to prove security. However, the security of dynamic assumptions like q-type assumptions is not well-understood, and the assumptions are often closely related to the scheme in which they are used. For example, the assumption may include a number of group elements that scales with the number of queries made by the adversary in the security proof. Further, it is known that many q-type assumptions become stronger as q grows [11], so we would ideally like to reduce security of ABE constructions to better understood assumptions of a static size, like the Decisional Linear Assumption (DLIN) or the Symmetric External Diffie-Hellman Assumption (SXDH).

A natural class of access policies one would like to be able to support in an ABE construction is that of general Boolean formulas. Unfortunately, it has proven tremendously difficult to construct efficient ABE for general Boolean formulas with adaptive security under static assumptions. All constructions except for [32] suffer from a "one-use restriction." That is, they only natively support read-once Boolean formulas, or formulas where attributes are used at most once in inputs. One way to extend such constructions to support formulas that use attributes more than once (say, k times) is to use k copies of new "meta-attributes" that stand for each use of the original attribute, and are handled as a group [21]. The downside of this approach is that it destroys the compactness of the construction – for KP-ABE, the size of the ciphertexts no longer depends on just the attribute set of the ciphertext, but also on the complexity of the formulas that the scheme supports (namely, the ciphertexts grow linearly with the maximum number of attribute uses in any formula supported). Ciphertexts associated with n' attributes in a scheme like [21] where policies can reuse attributes at most k times are of size $O(n' \cdot k)$.

Takashima presented the first KP-ABE scheme (proven adaptively secure from static assumptions) with ciphertexts that do not grow directly with the number of attribute uses [32], but unfortunately, the construction still has a dependence on the set of allowed policies. Specifically, ciphertexts are of size $O(n + r)$, where n is the maximum size of any supported attribute set and r is the maximum number of columns in any policy matrix supported (this is the policy dependency). For (fan-in 2) Boolean formulas, standard techniques [22] to translate the formula into a policy matrix result in r being equal to the number of AND gates in the formula. Additionally, this dependence on n, the maximum size of any supported attribute set rather than only the attribute set of the relevant ciphetext is undesirable, since one can imagine the size of each ciphertext's associated attribute set varying wildly from the worst-case maximum-sized set supported by the system. In fact, it is unclear whether $O(n + r)$-size ciphertexts are ever an asymptotic improvement over the $O(n' \cdot k)$-size ciphertexts of all other known ABE schemes proved adaptively secure under static assumptions.

reference	$\lvert sk\rvert$	$\lvert ct\rvert$	assumption
[21]	$O(\lvert f\rvert)$	$O(n' \cdot k)$	DLIN
[28]	$O(\lvert f\rvert)$	$O(n' \cdot k)$	DLIN
[9]	$O(\lvert f\rvert)$	$O(n' \cdot k)$	k-LIN
[10]	$O(\lvert f\rvert)$	$O(n' \cdot k)$	SXDH
[32]	$O(\lvert f\rvert)$	$O(n + r)$	DLIN
Ours	$O(\lvert f\rvert)$	$O(n' + r)$	SXDH

Fig. 1. Summary of several KP-ABE schemes proven adaptively secure under static assumptions for monotone span programs. Here, n' is the number of attributes associated to the ciphertext, n is the maximum size of any supported attribute set, r is the maximum number of columns in any policy matrix, and k is the maximum number of attribute reuses in any policy (except in the name for the "k-LIN" assumption, which is unrelated and an unfortunate overloading).

1.1 Our Result

In this work, we describe a KP-ABE construction that mitigates one of the two undesirable dependencies of [32], featuring ciphertexts of size $O(n' + r)$ instead of $O(n + r)$ (while remaining adaptively secure from a static assumption: the Symmetric Diffie-Hellman Assumption (SXDH) and allowing the reuse of attributes in its monotone span program policies). This significant improvement allows us to rigorously argue that there exist classes of access policies for which our construction enjoys an asymptotic improvement over the state of the art. We note that our construction is for the small-universe setting, where attributes come from a polynomial (in the security parameter) sized universe that is fixed upon setup, whereas the construction of [32] supports an attribute universe that may be exponentially large. This allows us to focus on the techniques required to asymptotically improve the ciphertext size. Our scheme is likely translatable to accommodate a large attribute universe without sacrificing asymptotic efficiency, but we leave this for future work.

Our construction avoids a dependence on k, the multiplicity of attribute-reuse in supported policies, but retains the dependence on r, the number of columns in supported policy matrices. We view reducing this last dependence to achieve truly compact adaptively secure ABE from a static assumption as an interesting open problem.

1.2 Comparing Perfomance

Figure 1 contains a comparison of several KP-ABE schemes proven adaptively secure under static assumptions for monotone span programs.

An obvious question in comparing our construction to the state-of-the-art is: how does r compare to k? Is $n' + r$ ever better than $n' \cdot k$? It is easy to come up with individual formulas where this is the case, but it's not obvious that such a formula can't always be "compressed" to an equivalent formula that has less attribute-reuse. In general, circuit/formula minimization questions like this are difficult to answer.

Fortunately, we can make a simple counting argument to show that indeed there are classes of functions which cannot be expressed using Boolean formulas with much smaller maximum attribute reuse than the maximum number of AND gates within the class. To see this, consider some subset of x attributes in the attribute universe. There are 2^{2^x} Boolean functions on these attributes, and we can express each function as a DNF in the naive way as a formula that uses at most $O(2^x)$ AND gates. So, for this class of functions, $r = O(2^x)$.

However, counting the number of different Boolean formulas that could attempt to realize these functions using a maximum k reuses of any attribute shows that at least $k = \Omega(2^x)$ attribute-reuses are required to realize all of the functions in this class. In this case, we see our construction enjoys a multiplicative to additive improvement (from $n' \cdot \Omega(2^x)$ to $n' + O(2^x)$).

1.3 Technical Details

Our construction can be seen as combining the best of both worlds between the construction of [32], which is the first to not directly depend on the number of attribute-reuses (while adaptively secure from a static assumption), and the lineage of [17,21,19,10], which enjoys ciphertexts that are independent of the size of the attribute universe (they depend only on the number of attributes actually associated with the ciphertext).

Specifically, all of these schemes are based on linear secret sharing and are built using bilinear groups. Given a matrix M representing a monotone span program, linear secret shares of α are constructed by choosing randomness r_i, then computing $M \cdot (\alpha, r_2, ..., r_m)$ to obtain a vector of shares $\boldsymbol{\lambda}$. The constructions of [17,21,19,10] embed these shares into their constructions' secret keys, where they are hidden by attribute-randomness that can only be removed using corresponding elements from a ciphertext. See Figure 2 for an example. A crucial step of the dual-system proof [33] of adaptive security occurs when secret shares in the dual "semi-functional" space of a secret key are changed from sharing 0 to sharing a random element α' (in [24], this is the change from "nominal semi-functional" to "temporary semi-functional"). This is the step of the proof that uses the fact that the keys requested by an adversary are not allowed to decrypt the challenge ciphertext, to argue that there exists different randomness r'_i where a sharing of 0 using the r_i randomness looks identically distributed to a sharing of random α' using the r'_i randomness, as long as the only shares seen are not allowed to reconstruct the secret. Crucially, the alternative randomness r'_i is not defined until the challenge ciphertext is requested (as the challenge ciphertext defines which shares in the key are allowed to be seen). The constructions in the [21] lineage therefore require that the change in the secret shares in their keys be information theoretic (so they can be implicitly changed upon challenge ciphertext creation). This turns out to be the root of the one-time attribute use restriction (reusing attributes prevents this information-theoretic argument from working).

[32] employs a technique of delayed share construction to get around this problem. Specifically, the construction does not construct a secret sharing $\{\lambda_j\}$

$$\{g^{\lambda_j + a_{\rho(j)} y_j}, \quad g^{y_j}\}_{j \in M}$$

Fig. 2. Example secret key

which is embedded in the secret key, but instead keeps the components that generate λ_j (vectors $\boldsymbol{M}_j$ and $(\alpha, r_2, ..., r_m)$) separate *until decryption*. The $\boldsymbol{M}_j$ portion is embedded in the key and the randomness $(\alpha, r_2, ..., r_m)$ is stored *in the ciphertext*. Decryption computes the dot product of these two components to implicitly construct λ_j that function in the same way as before. The advantage of this approach is that the randomness used in the secret shares is not needed until the challenge ciphertext is requested, so computational assumptions can be used to side-step the one-time attribute use restriction that comes with information-theoretic changes.

Like [21], [32] also masks secret key components, making them only available to ciphertexts associated with the appropriate attributes. However [32] does this via a somewhat blunt tool: namely, its secret keys contain a vector $\boldsymbol{y}$ which can encode orthogonality relationships with any subset of the attributes associated with a ciphertext and whose length is as large as the maximum attribute set supported by the system.

In contrast, the "share encapsulation" in [21] demonstrated in Figure 2 can be thought of as using a vector of dimension 2 to perform the same job. $(a_{\rho(j)} y_j, y_j)$ is being used to hide the share λ_j and share retrieval will be allowed only give a ciphertext with an "orthogonal" vector: $(s, -s a_{\rho(j)})$. Our construction can be seen as essentially replacing [32]'s vector $\boldsymbol{y}$ with constant-dimensioned vectors like this, resulting in a ciphertext dependent only on the number of attributes associated with it, just like all previous schemes. Essentially, an information-theoretic "encapsulation" argument supported by the vector $\boldsymbol{y}$ for all shares is replaced with a computational one using a vectors of a constant size for each attribute. Doing so makes the dual-system hybrid more delicate, as it requires careful management of rerandomization across the now greatly reduced dimensions.

1.4 Related Work

Additional work on ABE in the bilinear setting includes various constructions of KP-ABE and CP-ABE schemes (e.g. [5,30,16,18]), schemes supporting multiple authorities (e.g. [7,8,29,22]), and schemes supporting large attribute universes (e.g. [23,28,31,2,19,3,6,15,1,10]).

The construction of [14] supports circuit access policies rather than monotone span programs or Boolean formulas, which makes it more expressive than any known bilinear scheme. It was proven selectively secure under the standard LWE assumption. The construction of [6] later extended this to semi-adaptive security for circuit access policies from LWE. Proving full adaptive security for a ABE scheme supporting circuits from LWE or an assumption on bilinear maps is an interesting open problem.

Circuit policies are supported by the construction in [12] based on multilinear maps. This scheme is proven selectively secure, under a particular computational hardness assumption for multilinear groups. The multilinear scheme in [13] achieves adaptive security, relying on computational hardness assumptions in multilinear groups.

2 Preliminaries

We will write $a \leftarrow \mathbb{Z}_p$ to denote choosing a uniformly at random from set $\mathbb{Z}_p$ and will abuse notation to use $j \in M$ as a subscript to denote each index j of the rows M_j of matrix M.

2.1 Prime Order Bilinear Groups

We construct our system in prime order asymmetric bilinear groups. We let $\mathcal{G}$ denote a group generator - an algorithm which takes a security parameter λ as input and outputs (p, G, H, G_T, e), where p is a prime, G, H and G_T are cyclic groups of order p, and $e : G \times H \rightarrow G_T$ is a map with the following properties:

1. (Bilinear) $\forall g \in G, h \in H,\ a, b \in \mathbb{Z}_p,\ e(g^a, h^b) = e(g, h)^{ab}$
2. (Non-degenerate) $\exists g \in G, h \in H$ such that $e(g, h)$ has order p in G_T.

We refer to G, H as the *source groups* and G_T as the *target group*. We assume that the group operations in G, H and G_T and the map e are computable in polynomial time with respect to λ, and the group descriptions of G, H and G_T include a generator of each group.

2.2 Dual Pairing Vector Spaces

We will employ the concept of dual pairing vector spaces from [26,27], where we'll denote choosing random dual orthogonal bases as: $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathit{Dual}(\mathbb{Z}_p^n)$. Such bases are collections of linearly independent vectors chosen at random up to orthogonality constraints ($\boldsymbol{b}_i \cdot \boldsymbol{b}_i^* = 1, \boldsymbol{b}_i \cdot \boldsymbol{b}_j^* = 0$ for $i \neq j$). For example, one can implement $\mathit{Dual}(\mathbb{Z}_p^n$ by choosing a random invertible matrix B, setting $\mathbb{B} := B$ which then defines $\mathbb{B}^*$ as $\mathbb{B}^* := (B^{-1})^T$. Note that the dual basis generation procedure satisfies the property that, if R is an invertible matrix, then $(\mathbb{B}, \mathbb{B}^*)$ and $(R \cdot \mathbb{B}, (R^{-1})^T \cdot \mathbb{B}^*)$ are distributed identically when $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathit{Dual}(\mathbb{Z}_p^n)$. We will use this fact in our security proof to introduce new randomness into free dimensions of the construction as well as to embed computational assumptions. Finally, we will write $g^{\boldsymbol{v}}$ to denote the vector of group elements $(g^{v_1}, ..., g^{v_n})$, and will use the notation: $(x_1, ..., x_n)_{\mathbb{B}}$ to denote $g^{x_1 \boldsymbol{b}_1} \cdot ... \cdot g^{x_n \boldsymbol{b}_n}$.

2.3 Complexity Assumptions

The security of our system will be reduced to the Symmetric External Diffie-Hellman assumption (SXDH). We use the notation $x \leftarrow S$ to express that element x is chosen uniformly at random from the finite set S.

Symmetric External Diffie-Hellman Assumption (SXDH) The SXDH problem in G is stated as follows: given an asymmetric bilinear group (G, H) of prime order p with respective generators g, h, and given g^a, g^b and $T = g^{ab+r^*} \in G$ where $a, b \leftarrow \mathbb{Z}_p$ and either $r^* = 0$ or $r \leftarrow \mathbb{Z}_p$, output "yes" if r is a random element of $\mathbb{Z}_p$ and "no" otherwise. The SXDH problem in H is stated symmetrically, swapping the role of G and H.

Definition 1. *SXDH Assumption in (G, H): no polynomial time algorithm can achieve non-negligible advantage in deciding the SXDH problem in G or the SXDH problem in H.*

2.4 Background for ABE

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.

Monotone Span Programs / Linear Secret Sharing Schemes Our construction uses linear secret-sharing schemes (LSSS) to realize monotone span program access structures [25]. We use the following definition (adapted from [4]). In the context of ABE, attributes will play the role of parties and will be represented as indexes $i \in [|\mathcal{U}|]$ for a fixed universe $\mathcal{U}$.

Definition 2. *(Linear Secret-Sharing Schemes (LSSS)) A secret sharing scheme Π over a set of attributes is called* linear *(over $\mathbb{Z}_p$) if*

1. *The shares belonging to all attributes form a vector over $\mathbb{Z}_p$.*
2. *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 $i = \rho(j)$ (ρ is a mapping that maintains the relationship between matrix rows and attributes). 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 $(\Lambda v)_j = \lambda_j$ belongs to attribute $i = \rho(j)$.*

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 $\{\omega_i \in \mathbb{Z}_p\}_{i \in S^*}$ such that, for any valid shares $\{\lambda_i\}$ of a secret s according to Π, we have: $\sum_{i \in S^*} \omega_i \lambda_i = s$. These constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generating matrix Λ [4]. For unauthorized sets, no such S^*, $\{\omega_i\}$ exist.

For any set S of unauthorized shares, since the vector $(1, 0, ..., 0)$ is not in the span of rows indexed by S, then there is some vector $\boldsymbol{w}$ that is orthogonal to all of the rows of Λ indexed by S but is not orthogonal to $(1, 0, ..., 0)$. By scaling this vector, we can maintain these orthogonality relationships and force the first coordinate w_1 to be 1. Our proof of security will use the existence of this vector.

KP-ABE Definition A key-policy attribute-based encryption system consists of four algorithms: Setup, Encrypt, KeyGen, and Decrypt.

Setup($\lambda, \mathcal{U}) \rightarrow (\text{PP}, \text{MSK})$ The setup algorithm takes in the security parameter λ and the attribute universe description $\mathcal{U}$. It outputs the public parameters PP and a master secret key MSK.

Encrypt($\text{PP}, m, S) \rightarrow \text{CT}$ The encryption algorithm takes in the public parameters PP, the message m, and a set of attributes S. It will output a ciphertext CT. We assume that S is implicitly included in CT.

KeyGen$(\text{MSK}, \text{PP}, \mathbb{A}) \rightarrow \text{SK}$ The key generation algorithm takes in the master secret key MSK, the public parameters PP, and an access structure $\mathbb{A}$ over the universe of attributes. It outputs a private key SK which can be used to decrypt ciphertexts encrypted under a set of attributes which satisfies $\mathbb{A}$. We assume that $\mathbb{A}$ is implicitly included in SK.

Decrypt$(\text{PP}, \text{CT}, \text{SK}) \rightarrow m$ The decryption algorithm takes in the public parameters PP, a ciphertext CT encrypted under a set of attributes S, and a private key SK for an access structure $\mathbb{A}$. If the set of attributes of the ciphertext satisfies the access structure of the private key, it outputs the message m.

Adaptive Security for KP-ABE Systems We define adaptive security for KP-ABE Systems in terms of the following game:

Setup The challenger runs the Setup algorithm and gives the public parameters to the attacker.

Phase 1 The attacker queries the challenger for private keys corresponding to access structures.

Challenge The attacker declares two equal length messages M_0, M_1 and a set of attributes $A \subseteq \mathcal{U}$ where $\mathcal{U}$ is the attribute universe such that A does not satisfy the access structure of any of the keys requested in Phase 1. The challenger flips a random coin $\beta \in \{0, 1\}$, encrypts M_β under S to yield ciphertext CT_β and gives CT_β to the attacker.

Phase 2 The attacker queries the challenger for private keys corresponding to access structures that are not satisfied by S.

Guess The attacker outputs a guess β'.

Definition 3. *The advantage of an attacker $\mathcal{A}$ in this game is defined as* $Adv_{\mathcal{A}}^{KP-ABE}(\lambda) = \Pr[\beta = \beta'] - \frac{1}{2}$.

Definition 4. *A key-policy attribute based encryption scheme is adaptively secure if no polynomial time algorithm can achieve a non-negligible advantage in the above security game.*

3 Construction

Setup$(\lambda, \mathcal{U}) \rightarrow PP, MSK$ The setup algorithm chooses an asymmetric bilinear group $\mathcal{G}(\lambda) \rightarrow (p, G, H, G_T, e)$. It then chooses random generators $g \in G, h \in H$. For $i \in [k]$ where $k = |\mathcal{U}|$ it chooses values $a_i \leftarrow \mathbb{Z}_p$. It then generates random dual orthonormal sets:

$$(\mathbb{D}, \mathbb{D}^*) \leftarrow Dual(\mathbb{Z}_p^6)$$
$$(\mathbb{B}, \mathbb{B}^*) \leftarrow Dual(\mathbb{Z}_p^{3(r+1)})$$
$$(\mathbb{A}_i, \mathbb{A}_i^*) \leftarrow Dual(\mathbb{Z}_p^3) \text{ for } i \in [k]$$

The public parameters PP are:

$$e(g, h)$$
$$(\boldsymbol{e}_1)_{\mathbb{D}^*}, (\boldsymbol{e}_2)_{\mathbb{D}^*}$$
$$\{(\boldsymbol{e}_i)_{\mathbb{B}^*}\}_{i \in [r+1]}$$
$$\{(a_i, 0, 0)_{\mathbb{A}_i^*}\}_{i \in [k]}$$

The MSK is:

$$(\boldsymbol{e}_1)_{\mathbb{D}}, (\boldsymbol{e}_2)_{\mathbb{D}}$$
$$\{(\boldsymbol{e}_i)_{\mathbb{B}}\}_{i \in r+1}$$
$$\{(1, 0, 0)_{\mathbb{A}_i}\}_{i \in [k]}$$

Such a construction is equipped to create keys for access policies which include attributes $i \in \mathcal{U}$.

Encrypt$(m, S, PP) \rightarrow CT$ The encryption algorithm draws $\alpha, \Delta, s, z_i \leftarrow \mathbb{Z}_p$ (for $i \in [r]$) and forms the ciphertext as:

$$CT_S = (C_0, C_1, C_2, \{C_{3,i}\}_{i \in S})$$

where

$$C_0 := m \cdot e(g, h)^{\alpha}$$
$$C_1 := (\alpha, -\Delta, \mathbf{0}^2, \mathbf{0}^2)_{\mathbb{D}^*}$$
$$C_2 := (\Delta, z_2, ..., z_r, s, \mathbf{0}^{r+1}, \mathbf{0}^{r+1})_{\mathbb{B}^*}$$
$$C_{3,i} := (sa_i, 0, 0)_{\mathbb{A}_i^*}$$

(This implicitly includes S)

KeyGen$(MSK, M, PP) \rightarrow SK$ The key generation algorithm takes in the public parameters, master secret key, and LSSS access matrix M. It chooses a random exponent $x \leftarrow \mathbb{Z}_p$. For each row j (associated with attribute $\rho(j)$) in the policy matrix M, it chooses exponent $y_j \leftarrow \mathbb{Z}_p$ and outputs the secret key:

$$SK_M = (K_1, \{K_{2,j}, K_{3,j}\}_{j \in M})$$

where:

$$\begin{aligned} K_1 &:= (1, x, \mathbf{0}^2, \mathbf{0}^2)_{\mathbb{D}} \\ K_{2,j} &:= (\text{——}x\boldsymbol{M}_j\text{——}, a_{\rho(j)} y_j, \mathbf{0}^{r+1}, \mathbf{0}^{r+1})_{\mathbb{B}} \\ K_{3,j} &:= (-y_j, 0, 0)_{\mathbb{A}_{\rho(j)}} \end{aligned}$$

Decrypt$(CT_S, SK_M, PP) \to m$ Given ciphertext $CT_S = (C_0, C_1, C_2, \{C_{3,i}\}_{i \in S})$ and secret key $SK_M = (K_1, \{K_{2,j}, K_{3,j}\}_{j \in M})$, if S satisfies M, then there is a set S^* of policy row indices such that $j \in S^* \implies \rho(j) \in S$ and there exist efficiently computable constants ω_j such that $\sum_{j \in S^*} \omega_j M_j \cdot \boldsymbol{z} = \Delta$ (recall section 2.4). The decryption algorithm computes these ω_j and then computes:

$$B = \prod_{j \in S^*} e(C_2, K_{2,j})^{\omega_j} \cdot e(C_{3,\rho(j)}, K_{3,j})^{\omega_j}$$

$$D = e(C_1, K_1)$$

and finally, computes and outputs:

$$\frac{C_0}{B \cdot D} = m$$

4 Correctness

This scheme satisfies correctness since:

$$\begin{aligned} B &= \prod_{j \in S^*} e(C_2, K_{2,j})^{\omega_j} \cdot e(C_{3,\rho(j)}, K_{3,j})^{\omega_j} \\ &= \prod_{j \in S^*} e\left(\begin{matrix} (\Delta, z_2, ..., z_r, \quad s, \quad \mathbf{0}^{r+1}, \mathbf{0}^{r+1})_{\mathbb{B}^*}, \\ (\text{——}x\boldsymbol{M}_j\text{——}, \ a_{\rho(j)} y_j, \mathbf{0}^{r+1}, \mathbf{0}^{r+1})_{\mathbb{B}} \end{matrix} \right)^{\omega_j} \cdot e\left(\begin{matrix} (\ sa_{\rho(j)}, 0, 0)_{\mathbb{A}^*_{\rho(j)}}, \\ (-y_j, 0, 0)_{\mathbb{A}_{\rho(j)}} \end{matrix} \right)^{\omega_j} \\ &= \prod_{j \in S^*} e(g, h)^{x\omega_j \lambda_j + s\omega_j a_{\rho(j)} y_j} \cdot e(g, h)^{-s\omega_j a_{\rho(j)} y_j} \\ &= e(g, h)^{x \sum_{j \in S^*} \omega_j \lambda_j} \\ &= e(g, h)^{x\Delta} \end{aligned}$$

$$\begin{aligned} D &= e(C_1, K_1) \\ &= e\left(\begin{matrix} (\alpha, -\Delta, \mathbf{0}^2, \mathbf{0}^2)_{\mathbb{D}^*} \\ (1, \quad x, \ \mathbf{0}^2, \mathbf{0}^2)_{\mathbb{D}} \end{matrix} \right) \\ &= e(g, h)^{\alpha - x\Delta} \end{aligned}$$

and finally:

$$\begin{aligned}\frac{C_0}{B \cdot D} &= \frac{m \cdot e(g,h)^{\alpha}}{e(g,h)^{x\Delta} \cdot e(g,h)^{\alpha - x\Delta}} \\ &= m\end{aligned}$$

5 Proof of Security

Our proof of security will consist of a hybrid sequence of games where the keys and challenge ciphertext are constructed according to various types. At a high level, the proof follows a typical dual-system hybrid structure, where the challenge ciphertext is first made "semi-functional," then the hybrid continues over the secret keys requested, transforming each key into a "semifunctional" variant which is useless to the attacker relative to the challenge (semifunctional) ciphertext.

There are two parts to the key hybrid: one that makes semifunctional keys which were requested before the challenge ciphertext and another that makes semifunctional keys which were requested after the challenge ciphertext. The high level reason for this difference is that for keys requested after the challenge ciphertext, the challenge attribute set is already known. This makes it easy to follow a standard selective security argument to make each key semifunctional. The harder part of the hybrid deals with making keys requested before the challenge ciphertext (and the challenge attribute set) is known. This is where we use the delayed randomness contained within our ciphertext as well as the fact that we allow the semifunctional ciphertext distributions to depend on the current key of the hybrid. This bifurcated approach to handling secret keys in a dual-system proof was first employed in [24] and later refined by [2,3]

A key step in our proof (and of [32]) is a lemma where each policy matrix row is isolated in turn against the ciphertext's $\boldsymbol{w}$ alternative randomness component and their dot product's distribution is used to argue that the row can be multiplied by an uncorrelated x^*. In [32], this argument takes advantage of the inefficient $\boldsymbol{y}$ vector, but for us, we need to delicately thread just enough randomness through the single attribute element a_i hiding each row to accomplish the same feat.

Theorem 1. *Under the SXDH assumption, our KP-ABE construction is adaptively secure against any polynomial time adversary $\mathcal{A}$.*

We give the proof of Theorem 1 in the full version of this paper [20].

6 Acknowledgements

This work was supported in part by The Leona M. & Harry B. Helmsley Charitable Trust; NSF grant CCF-1423306; and the Defense Advanced Research Project Agency (DARPA) and Army Research Office (ARO) under Contract W911NF-15-C-0236. The first author is additionally supported in part by an NSF Graduate

Research Fellowship DGE-16-44869. Any opinions, findings and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the the Defense Advanced Research Projects Agency, Army Research Office, the National Science Foundation, or the U.S. Government.

References

1. Agrawal, S., Chase, M.: Fame: Fast attribute-based message encryption. In: CCS (2017)
2. Attrapadung, N.: Dual system encryption via doubly selective security: Frame- work, fully secure functional encryption for regular languages, and more. In: EUROCRYPT. p. 57577 (2014)
3. Attrapadung, N.: Dual system encryption framework in prime-order groups via computational pair encodings. In: ASIACRYPT. p. 91623 (2016)
4. Beimel, A.: Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
5. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: Proceedings of the IEEE Symposium on Security and Privacy. pp. 321–334 (2007)
6. Brakerski, Z., Vaikuntanathan, V.: Circuit-abe from lwe: Unbounded attributes and semi-adaptive security. In: CRYPTO. pp. 363–384 (2016)
7. Chase, M.: Multi-authority attribute based encryption. In: Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings. pp. 515–534 (2007)
8. Chase, M., Chow, S.S.M.: Improving privacy and security in multi-authority attribute-based encryption. In: Proceedings of the 2009 ACM Conference on Computer and Communications Security. pp. 121–130 (2009)
9. Chen, J., Gay, R., Wee, H.: Improved dual system abe in prime-order groups via predicate encodings. In: EUROCRYPT. pp. 595–624 (2015)
10. Chen, J., Gong, J., Kowalczyk, L., Wee, H.: Unbounded abe via bilinear entropy expansion, revisited. In: EUROCRYPT. pp. 503–534 (2018)
11. Cheon, J.H.: Security analysis of the strong diffie-hellman problem. In: EUROCRYPT. pp. 1–11 (2006)
12. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: CRYPTO. pp. 479–499 (2013)
13. Garg, S., Gentry, C., Halevi, S., Zhandry, M.: Fully secure attribute based encryption from multilinear maps. IACR Cryptology ePrint Archive **2014**, 622 (2014), http://eprint.iacr.org/2014/622
14. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: STOC. pp. 545–554 (2013)
15. Goyal, R., Koppula, V., Waters, B.: Semi-adaptive security and bundling functionalities made generic and easy. In: TCC. pp. 361–388 (2016b)
16. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute-based encryption. In: ICALP (2008)
17. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute based encryption for fine-grained access control of encrypted data. In: ACM conference on Computer and Communications Security. pp. 89–98 (2006)
18. Jie, C., Wee, H.: Semi-adaptive attribute-based encryption and improved delegation for boolean formula. In: SCN. pp. 277–297 (2014)
19. Kowalczyk, L., Lewko, A.B.: Bilinear entropy expansion from the decisional linear assumption. In: CRYPTO. pp. 524–541 (2015)
20. Kowalczyk, L., Liu, J., Malkin, T., Meiyappan, K.: Mitigating the one-use restriction in attribute-based encryption. IACR Cryptology ePrint Archive **2018**, 645 (2018), https://eprint.iacr.org/2018/645.pdf

21. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: EUROCRYPT. pp. 62–91 (2010)
22. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: EUROCRYPT. pp. 568–588 (2011)
23. Lewko, A., Waters, B.: Unbounded hibe and attribute-based encryption. In: EUROCRYPT. pp. 547–567 (2011)
24. Lewko, A.B., Waters, B.: New proof methods for attribute-based encryption: Achieving full security through selective techniques. In: CRYPTO. pp. 180–198 (2012)
25. M.Karchmer, Wigderson, A.: On span programs. In: CCC. pp. 102–111 (1993)
26. Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. In: Pairing. pp. 57–74 (2008)
27. Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: ASIACRYPT. pp. 214–231 (2009)
28. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: ASIACRYPT. pp. 349–366 (2012)
29. Okamoto, T., Takashima, K.: Decentralized attribute-based signatures. In: PKC. pp. 125–142 (2013)
30. Ostrovksy, R., Sahai, A., Waters, B.: Attribute based encryption with non-monotonic access structures. In: ACM conference on Computer and Communications Security. pp. 195–203 (2007)
31. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: 2013 ACM Conference on Computer and Communications Security. pp. 463–474 (2013)
32. Takashima, K.: New proof techniques for DLIN-based adaptively secure attribute-based encryption. In: ACISP. pp. 85–105 (2017)
33. Waters, B.: Dual system encryption: realizing fully secure ibe and hibe under simple assumptions. In: CRYPTO. pp. 619–636 (2009)
34. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: PKC. pp. 53–70 (2011)