FABEO: Fast Attribute-Based Encryption with Optimal Security

Doreen Riepel\textsuperscript{1} and Hoeteck Wee\textsuperscript{2}

\textsuperscript{1} Ruhr-Universität Bochum, Germany
doreen.riepel@rub.de
\textsuperscript{2} NTT Research, Sunnyvale, USA
wee@di.ens.fr

Abstract. Attribute-based encryption (ABE) enables fine-grained access control on encrypted data and has a large number of practical applications. This paper presents FABEO: faster pairing-based ciphertext-policy and key-policy ABE schemes that support expressive policies and put no restriction on policy type or attributes, and the first to achieve optimal, adaptive security with multiple challenge ciphertexts. We implement our schemes and demonstrate that they perform better than the state-of-the-art (Bethencourt et al. S&P 2007, Agrawal et al., CCS 2017 and Ambrota et al., CCS 2017) on all parameters of practical interest.

Keywords: attribute-based encryption; generic group model; tightness

1 Introduction

Attribute-based encryption (ABE)\cite{ABEintro} extends classical public-key encryption to support fine-grained access control on encrypted data. ABE has applications in a variety of settings including electronic medical records\cite{ABEMedRec}, messaging systems\cite{ABEMsgSys}, online social networks\cite{ABESocNet} and information-centric networking\cite{ABEInfNet}. Companies like Cloudflare already use ABE to distribute private key storage across data centers\cite{ABECloudflare}.

ABE comes in two variants: ciphertext-policy (CP-ABE) and key-policy (KP-ABE), depending on whether access policies are attached to ciphertexts or to keys\cite{ABECryptoSurvey}. In CP-ABE, keys are associated with sets of attributes, and a key is able to recover the message hidden in a ciphertext if and only if the set of attributes satisfy the access policy attached to the ciphertext. For instance, a policy P could say ‘(Zipcode:90210 OR City:BeverlyHills) AND (AgeGroup:18-25)’ and an individual A could have a key for Zipcode:90210, AgeGroup:Over65, in which case A would not be able to decrypt any message encrypted under P. A KP-ABE is the dual of CP-ABE with ciphertexts attached to attribute sets and keys associated with access policies.

There is by now a vast body of research on ABE realizing a broad spectrum of trade-offs between efficiency, expressiveness, security and hardness assumptions. The state of the art for practical ABE schemes are encapsulated by the following pairing-based schemes: (i) BSW CP-ABE scheme (Bethencourt, Sahai and Waters\cite{BSW07}), (ii) FAME CP-ABE and KP-ABE schemes (Agrawal and Chase\cite{FAME}), and (iii) ABGW CP-ABE and KP-ABE schemes (Agrawal, Barthe, Gay and Wee\cite{ABGW}). These schemes simultaneously achieve the following properties that are highly desirable in practice:

(1) support expressive policies described by boolean formula and monotone span programs (MSP);
(2) put no restriction on size of policies or attribute sets;
(3) allow any arbitrary string such as street addresses to be used as an attribute;
(4) achieve the strong and natural notion of adaptive security, the defacto standard for ABE.

However, these schemes achieve incomparable efficiency guarantees, and deciding which one to deploy requires making complex performance trade-offs that depend on the policies that arise in the specific context.
### Scheme Comparison

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Unrestricted policies</th>
<th>Arbitrary attributes</th>
<th>Fast decryption</th>
<th>Attribute multi-use</th>
<th>Security bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-ABE BSW</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>$t^3/p$</td>
</tr>
<tr>
<td>CP-ABE Waters</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>–</td>
</tr>
<tr>
<td>CP-ABE FAME</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>$t^4/p$</td>
</tr>
<tr>
<td>Our FABEO [Fig 1]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>$t^2/p$</td>
</tr>
<tr>
<td>KP-ABE GPSW</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>–</td>
</tr>
<tr>
<td>KP-ABE ABGW</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>–</td>
</tr>
<tr>
<td>KP-ABE FAME</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>–</td>
</tr>
<tr>
<td>Our FABEO [Fig 1]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>$t^2/p$</td>
</tr>
</tbody>
</table>

Table 1: A property-wise comparison of the various ABE schemes we consider. The BSW, Waters and GPSW schemes were specified using symmetric pairings in the original works; throughout, we refer to the asymmetric variants from [2]. The last column shows the security bounds on the adversary’s advantage in the multi-ciphertext setting for the adaptively secure schemes, with dashes indicating selectively secure schemes.

### 1.1 Our Contributions

We present FABEO, new pairing-based KP-ABE and CP-ABE schemes achieving properties (1) – (4), with improved efficiency and quantitatively stronger security guarantees. FABEO uses asymmetric (Type-III) prime-order bilinear groups $\left(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\right)$ which support efficient hashing to $\mathbb{G}_1$. Ciphertexts and secret keys in FABEO comprise mostly of elements in the smaller and faster group $\mathbb{G}_1$, plus 1 or 2 elements in $\mathbb{G}_2$. Computation for key generation, encryption and decryption are mostly carried out in $\mathbb{G}_1$, with 2 to 3 pairings for decryption. We prove optimal security bounds for FABEO against adversaries that get an arbitrary number of ciphertexts and keys: in particular, when instantiated over the popular BLS12-381 curve, FABEO achieves close to 128-bit security.

FABEO subsumes BSW, FAME and ABGW on all parameters of practical interest. We improve upon the ciphertext and key sizes of all three schemes, as well as the running times. In particular, our ciphertexts are 66% smaller; encryption is (at least) 33% faster; and decryption uses fewer pairings. FABEO also supports multi-use of attributes like in BSW and ABGW (without an a-prior bound during set-up), with a small additive overhead in the multi-use parameter. See Table 1 for a property-wise comparison of our schemes against BSW, FAME, ABGW and other prominent schemes in the literature, as well as Tables 3 and 5 for a theoretical analysis and comparison for efficiency.

FABEO achieves properties (2) and (3) by hashing attributes to $\mathbb{G}_1$; smaller ciphertext/key sizes and fast decryption via randomness reuse (in CP-ABE ciphertexts and KP-ABE keys); and adaptive security without efficiency penalties by considering “generic” adversaries, a widely accepted model that captures all known attacks. While each of these techniques is already present in BSW, FAME, and ABGW and prior works, FABEO is the first to combine them in a single design, along with a novel analysis establishing optimal security.

**Optimal security.** We prove security of our schemes in the generic bilinear group model (GGM) [14,15,16] (as with BSW and ABGW), where we model the underlying hash function as a random oracle [11] (as with BSW and FAME). We show that any generic, adaptive adversary running in time $t$ and sees at most $t$ ciphertexts and keys breaks our schemes with probability at most $O(t^2/p)$, where $p$ is the order of the underlying group. This bound is optimal, since an adversary can break discrete log with the same probability. Prior ABE schemes, including BSW, FAME, and ABGW, achieve a bound of $O(t^3/p)$ or worse, since the security proofs only consider a single challenge ciphertext, and a hybrid argument is needed to achieve multi-ciphertext security.

**Proof framework and application.** In both our CP-ABE and KP-ABE schemes, the ciphertext $ct$ for $x$ and secret key $sk$ for $y$ are of the form:

$$ct = \left(g_1^{c_1(x, b)}, g_2^{c_2(x)}, e(g_1, g_2)^{c_1(x, b) s_1} \cdot M\right), sk = \left(g_1^{k_1(x, b, r)}, g_2^{k_2(x, b, r)}\right).$$
Here, \( s = (s_1, \ldots) \) and \( r \) are fresh randomness; \( g_1^b \) contains the hash of every attribute in the universe\(^3\) and \( c_y^1, c_y^2, k_y^1, k_y^2 \) are simple functions of degree 1 or 2. Roughly speaking, we show that for schemes of this form\(^4\) security for a single ciphertext-key pair implies optimal, adaptive security against generic adversaries with an arbitrary number of ciphertexts and keys. Our modular proof framework extends and generalizes an analogous statement shown in ABGW in several ways: (i) we allow \( c_y^2(s) \) to have arbitrary length instead of length 1, as is necessary to capture our CP-ABE scheme and the one below, (ii) we consider security with multiple ciphertexts, and (iii) we achieve optimal security.

Next, we describe an additional application of our proof framework that pertains to property (1). A limitation of boolean formula and monotone span programs is they do not capture computation over data of arbitrary, unbounded size, which arise settings such as genome sequencing, processing network and event logs, tax returns and virus scanners; such computation are better captured by regular languages, or deterministic finite automata (DFA). As a secondary contribution, we prove that Waters’ KP-ABE scheme for DFA \(^5\) achieves optimal, adaptive security. In this scheme, \( c_y^2(s) \) has arbitrary length that grows with \( x \). Compared to prior adaptively secure KP-ABE for DFA \(^7\)\(^3\)\(^4\)\(^2\)\(^9\), we obtain (at least) a 50% improvement in ciphertext and key sizes as well as running times.

Implementation and evaluation. We implement FABEO in the Charm framework \(^4\). Our experiments validate our theoretical analysis in Table 3 showing that FABEO improves on the performance of BSW, FAME and ABGW, for all of key generation, encryption and decryption. FABEO compares favorably even against the Waters CP-ABE \(^6\) and GSW KP-ABE \(^8\), even though these schemes do not achieve property (3). See Figure 3 in Section 7 for the performance of the algorithms of each scheme under various test cases. Our code is available on GitHub \(^9\).

All computations are performed on an ordinary laptop and we achieve practical results, even for large attribute sets and policies. Specifically for our CP-ABE with the MNT224 curve, set-up takes less than 0.02s, and it takes around 0.09s to generate a key for 100 attributes, and 0.18s to encrypt data under a policy that requires all 100 attributes. Decryption then takes only 0.02s. As a comparison, the ABGW CP-ABE scheme takes 0.63s to generate a key for the same number of attributes, 0.33s to encrypt and 0.48s to decrypt. In FAME, decryption takes 0.03s, and key generation and encryption are slower than ABGW.

Summary of Contributions. To summarize, our contributions are as follows:

- We present new KP-ABE and CP-ABE schemes for MSP with improved efficiency guarantees and the first to achieve optimal, adaptive security with multiple challenge ciphertexts.
- We provide a more general and modular framework for proving optimal ABE security in the GGM.
- We implement our KP-ABE and CP-ABE schemes for MSP and evaluate their performance for various parameters.
- We present and implement a new KP-ABE for DFA with optimal, adaptive security in the GGM.

1.2 Discussion and Related Work

We discuss additional context and related works.

Choosing curve parameters. When choosing curve parameters for a pairing-based scheme, practitioners often base the decisions on the hardness of the discrete log problem, and ignore the security bounds provided in security proof for the scheme. This is in part due to the limited number of pairing-friendly curves that are available in practice \(^12\), and the possibly prohibitive performance penalty from using a curve with larger bit security. In particular, there is an implicit expectation that a scheme instantiated over a curve with 128-bit security should also achieve close to 128-bit security. Our work takes a step towards rigorously justifying this expectation in the context of pairing-based ABE.

\(^3\) Ignoring for now the fact that \( b \) has exponential length.

\(^4\) The ABGW CP-ABE and KP-ABE schemes we compare with are not of this form since the \( k_y^1 \) computes a rational function.

\(^5\) Waters only proved weaker, selective security for his scheme. More precisely, we consider a variant of Waters’ scheme with smaller keys from \(^28\).
Fig. 1: Our CP-ABE (left) and KP-ABE (right) scheme for monotone span programs \((M \in \mathbb{Z}_p^{n_1 \times n_2}, \pi : [n_1] \to \mathcal{U})\). We define \(\rho(i) := |\{z \mid \pi(z) = \pi(i), z \leq i\}|\) and \(\tau = \max_{i \in [n_1]} \rho(i)\) corresponding to maximum number of times an attribute is used in \(M\).

Achieving adaptive security. There are two main approaches for realizing adaptive security for ABE schemes in the literature: (1) prove security against generic adversaries as was done in BSW, ABGW and this work, and (2) adopt the dual system encryption framework as used in FAME, which allows us to base security on SXDH and DLIN (and in some settings, with the additional use of \(q\)-type assumptions). While the latter yields theoretically stronger results, it incurs a huge penalty in efficiency: for security from \(k\)-LIN \((k = 1 \text{ corresponds to SXDH and } k = 2 \text{ to DLIN})\), it requires \((\text{at least})\) a factor \(k + 1\) blow-up in ciphertext and key sizes as well as running times for encryption, key generation and decryption \([10,38]\). Moreover, the schemes have a more complex structure, and the security proofs are also substantially more complex. Another drawback is that the proofs typically require a hybrid argument over the keys and the ciphertexts, so we cannot hope for a security bound better than \(O(t^2/p)\).

GGM security. We argue that GGM security is sufficient for most practical applications. The reasoning is two-fold: First, our understanding of pairing curves has advanced substantially over the past two decades, with increasing adoption (e.g. Cloudflare and ZCash) as well as on-going standardization \([17]\). The known attacks fall broadly into two categories: (1) attacks on discrete log, most notably the exTNFS in \([36]\), rendering the curves unsuitable for any applications, (2) attacks that are captured by the GGM \([19]\). In short, there is in practice no discernible distinction between the standard assumptions like SXDH and GGM security. Second, it is much easier to break a real-world system via side channel attacks or poor security practices (e.g. phishing attacks or weak passwords) than to come up with an attack outside of the GGM. Indeed, a large number of recent works also use the GGM to analyze practical cryptosystems, e.g. \([31,10,32]\).

Optimal and tight security. This work falls under a broader cryptographic research agenda of achieving optimal security and tight security reductions, for instance, recent works on symmetric-key encryption \([33]\), signature schemes \([21]\) and TLS 1.3 \([26,22]\). In the context of ABE, optimal security was only
previously known in very limited settings, namely identity-based encryption and its hierarchical variant [25][13][24][27][15][17]: we clarify that these works focus on the more challenging goal of basing security on static assumptions such as DLIN. In particular, these are the only settings where we know how to carry out a dual system encryption proof with security bound better than $O(t^4/p)$.

Benchmarking ABE schemes. Two very recent works [14][20] looked into benchmarking pairing-based ABE schemes, focusing on low-level optimizations (whereas our work focuses on high-level design as well as new security guarantees and proof techniques): the first for CP-ABE covering BSW, Waters, and FAME but not ABGW, and the second for broadcast encryption. Both works highlight the complexity of effective benchmarking due to incomparable trade-offs between efficiency, expressiveness, security and hardness assumptions, which we alluded to at the beginning of the paper. Our results, together with those in ABGW and the preceding discussion, support the thesis that one should consider GGM-based schemes for benchmarking, since we can achieve the strongest notion of adaptive security without efficiency penalties.

1.3 Technical Overview

Let $(G_1, G_2, G_T)$ be an (asymmetric) bilinear group of prime order $p$, along with a pairing $e : G_1 \times G_2 \rightarrow G_T$ and generators $g_1, g_2$ for $G_1, G_2$ respectively. In general, the bit sizes of group elements in $G_2$ are 2-3 times that of $G_1$ and group operations in $G_2$ take (at least) twice as much time. In addition, we can securely hash into $G_1$ at the cost of roughly one exponentiation in $G_1$.

High-level design. We begin with a high-level overview of our KP-ABE scheme described in Figure 1. An MSP is given by a matrix $M$ and a function $\pi$ that maps each row of $M$ to an attribute. Following [23], we design the ciphertexts and secret keys so that for each row $i$ in $M$ such that $\pi(i)$ appears in the attribute set, decryption will compute

$$e(g_1, g_2)^{s_i \alpha_i}$$

where $\alpha_i$ is a share of the master secret key $\alpha$ and $s_i \leftarrow \mathbb{Z}_p$ is the encryption randomness. The values in $\mathbb{G}_1$ can then be combined to recover the blinding factor $e(g_1, g_2)^{s_i \alpha_i}$.

To realize the above invariant, we compute

$$(g_2^{\alpha_i}, H(\pi(i))^{s_i}) \in ct, \quad (g_2^{\alpha_i}, g_1^{s_r}H(\pi(i))^r) \in sk$$

so that we can compute (1) using $e(g_1^{\alpha_i}H(\pi(i))^r, g_2^{s_i})/e(H(\pi(i))^{s_i}, g_2^{s_r})$. If $\pi$ is injective (i.e., no attribute multi-use), we can use the same $r$ across all the rows in $M$, to keep the key size small. This way, we can also carry out decryption using two pairings and with most of the computation in the faster group $G_1$. So far, the scheme is the same as the one in [23]. We further extend [24] to support attribute multi-use as follows: we sample $r_1, \ldots, r_r$, where $r$ is maximum number of times an attribute is used in $M$, and we replace $r$ in $(g_2^{\alpha_i}, g_1^{s_r}H(\pi(i))^r) \in sk$ with $r_{\rho(i)}$, corresponding to the $\rho(i)$th occurrence of attribute $\pi(i)$ in $M$. The key size increases by $\tau - 1$ group elements, and decryption now requires $\tau + 1$ pairings. In contrast,

- BSW uses a different $r_i$ for each row in $M$, namely $(g_2^{\alpha_i}, g_1^{s_r}H(\pi(i))^r) \in sk$ (here, we are describing the KP-ABE analogue of the BSW CP-ABE). This incurs a multiplicative factor 2 blow-up in key size, and decryption requires computing a pairing for each row of $M$.

- ABGW uses

$$g_2^{s_i - s_j}, g_1^{s_i(b_1 + \pi(i) b_2)} \in ct, \quad g_2^{s_i - s_j}, g_1^{s_i(b_1 + \pi(i) b_2)} \in sk$$

where $g_1^{b_1}, g_1^{b_2}$ comes from $mpk$. This incurs (at least) a factor 2 blow-up in ciphertext and key sizes, and an extra exponentiation per attribute during encryption. Decryption requires computing a pairing for each attribute.

\textsuperscript{6} by writing $\prod_i (e(g_1, g_2)^{s_i \alpha_i})^{-s_i}$ as $e(\prod_i (g_1^{s_i}H(\pi(i))^r), g_2^{s_i}) \cdot e(\prod_i (H(\pi(i))^s_i), g_2^{s_i})^{-1}$.
FAME replaces $g_2^{s_1}, g_2^{s_2}$ with DLIN-tuples in $\mathbb{G}_2$ in order to achieve security under the DLIN assumption using the dual system encryption framework as described in Section 1.2. Overall, this incurs a factor 3 blow-up in ciphertext and key sizes, as well as a factor 3-6 blow-up in running time for encryption and key generation.

Our CP-ABE scheme is conceptually the dual of our KP-ABE, though algebraically more intricate and less intuitive (the same holds for BSW, FAME, and ABGW). Briefly, instead of (1), decryption computes $e(g_1, g_2)^{g_1 r}$ where $g_1$ is a share of the encryption randomness $s_1$; $g_2^{s_1}$ is specified in the public key; and $r$ comes from key generation randomness. These values can then be combined to compute $e(g_1, g_2)^{g_1 r}$, which is in turn used to recover the blinding factor $e(g_1, g_2)^{a \cdot r}$. Our CP-ABE scheme is the same as the AC17-LU-OK and AC17-LU-CP schemes in the independent work [41], which asserts selective security under $q$-type assumptions without a formal security proof.

**Proof strategy.** We provide a unified proof security of our KP-ABE and CP-ABE schemes in the GGM, where we model $H$ as a random oracle. At a high level, we follow the framework in [9]. Both our KP-ABE and CP-ABE schemes have the following structure where the ciphertext is associated with a label $x$ and the secret key with a label $y$ ($x$ is an attribute set for KP-ABE and a policy for CP-ABE, and vice-versa for $y$)

$$ct_x = \left( g_1^{c_1(s \otimes b)}, g_2^{c_2(s)}, e(g_1, g_2)^{a \cdot s_1}, M \right), \ sk_y = \left( g_1^{k_1(\alpha, r, b \otimes r)}, g_2^{k_2(r)} \right),$$

where

- $s = (s_1, \ldots)$ and $r$ are random vectors over $Z_p$ corresponding to randomness for encryption and key generation;
- $g_1^{b}$ contains the hash of every attribute in the universe, along with $g_1^{b'}$ for our CP-ABE (note that the length of $b$ is exponential, but the $c_1, k_1$ only depend on a polynomial number of entries of $b$);
- $c_1, c_2, k_1, k_2$ are linear functions over $Z_p$ (therefore $c_1$ and $k_1$ computes degree 2 functions of $s, b, r, \alpha$);
- decryption uses the pairing to compute $e(g_1, g_2)^{c_1(s \otimes b) \cdot k_1 r} \cdot e(g_1, g_2)^{k_1(\alpha, r, b \otimes r) \cdot c_2(s)}$, followed by additional linear computation in the exponent to recover the blinding factor $e(g_1, g_2)^{a \cdot s_1}$.

We refer to ABE schemes with the above structure as a PES-ABE (PES is short for pair encoding schemes [7]). Towards proving GGM security, we consider notions of symbolic security for PES-ABE, where an adversary sees abstract expressions for group elements in the form of polynomials. The proof of security of our KP-ABE and CP-ABE schemes follows the following modular framework:

**Step 1.** We show that our KP-ABE and CP-ABE schemes satisfy the syntax of a PES-ABE and (1,1) symbolic security, a relaxation of ABE security where the adversary is selective and only receives a single ciphertext and single secret key.

**Step 2.** We prove that any PES-ABE satisfying (1,1) symbolic security also satisfies strong symbolic security, where the adversary is still selective but can see the public key as well as an arbitrary number of ciphertexts and secret keys.

**Step 3.** We prove that any PES-ABE satisfying strong symbolic security is adaptively secure in the GGM with optimal security.

We now describe the key differences between our framework and the one in ABGW:

- The syntax for PES-ABE is different: (i) both $ct_x$ and $sk_y$ contain elements from both $\mathbb{G}_1$ and $\mathbb{G}_2$, and (ii) we generate $g_1^{b}$ using a random oracle.

- We introduce a strengthening of (1,1) symbolic security where we essentially require that all of $[\alpha c_1(s)]_T$ are pseudorandom, and not just $[\alpha s_1]_T$. Our notion is also weaker in that the proof only needs to reason about the terms $e(g_1, g_2)^{c_1(s \otimes b) \cdot k_1 r} \cdot e(g_1, g_2)^{k_1(\alpha, r, b \otimes r) \cdot c_2(s)}$.

7 The tensor product $u \otimes v$ of two vectors $u = (u_1, u_2, \ldots)$ and $v = (v_1, v_2, \ldots)$ is a vector $(u_1 v_1, u_1 v_2, \ldots)$ containing all pairwise products of the entries in $u$ and $v$.

8 In this overview, we use selective to refer to an adversary that specifies all of its ciphertext and key queries in advance.
The ABGW KP-ABE and CP-ABE schemes for MSP do not satisfy the syntax of a PES-ABE since $k^2_y$ computes rational functions with linear functions $b$ in the denominator and therefore proving security of these schemes require directly establishing strong symbolic security;

- The analogue of strong symbolic security in ABGW in Steps 2 and 3 considers only a single challenge ciphertext.
- We achieve a security bound of $O(t^2/p)$ in Step 3, whereas ABGW achieves $O(t^3/p)$. Our proof crucially relies on the fact that $e_1$, $e_2$, $k^1_y$, $k^2_y$ compute functions of degree at most 2 in the inputs so that we only need to apply Schwartz-Zippel to constant-degree polynomials. The proof in ABGW applies Schwartz-Zippel to polynomials of degree $t$ in order to “clear the denominators” across $t$ keys.

2 Preliminaries

We will first fix some notation that we will use throughout the paper. For integers $m$, $n$ where $m < n$, $[m, n]$ denotes the set $m, m + 1, ..., n$. For $m = 1$, we simply write $[n]$. For a prime $p$, let $\mathbb{Z}_p$ denote the set $[0, p - 1]$, where addition and multiplication are computed modulo $p$. For a set $\mathcal{S}$, $s \leftarrow \mathcal{S}$ denotes that $s$ is sampled uniformly and independently at random from $\mathcal{S}$. $y \leftarrow \mathcal{A}(x_1, x_2, \ldots)$ denotes that on input $x_1, x_2, \ldots$ the probabilistic algorithm $\mathcal{A}$ returns $y$. $\mathcal{A}^\theta$ denotes that algorithm $\mathcal{A}$ has access to oracle $\mathcal{O}$. An adversary is a probabilistic algorithm. A probabilistic algorithm is called efficient or PPT if its running time is bounded by some polynomial in the length of its input.

We use lower case bold-face letters for row vectors, where $\|\|$ denotes concatenation of row vectors. $v[i]$ denotes the $i$-th coordinate of the vector $v$. Given a vector $v$ of polynomials of length $m$ over $\mathbb{Z}_p$, we write $\text{span}(v)$ to denote $\{v \cdot e^t : e \in \mathbb{Z}_m\}$. Formal variables are marked with a tilde. We write $\tilde{v} \leftarrow \text{Var}^n$ to pick $n$ formal variables.

2.1 Pairing Groups

Let $\text{GroupGen}$ be a PPT algorithm that takes a security parameter $1^\lambda$ as input and returns a group description $G := (p, G_1, G_2, G_T, e, g_1, g_2)$, where $p$ is a prime of $\Theta(\lambda)$ bits, $G_1$, $G_2$ and $G_T$ are cyclic groups of order $p$, $e : G_1 \times G_2 \rightarrow G_T$ is a non-degenerate bilinear map (also called pairing) and $g_1$ resp. $g_2$ or generators of $G_1$ resp. $G_2$. The generator $g_T$ of $G_T$ can be computed as $e(g_1, g_2)$. We require that the group operations in $G_1$, $G_2$, $G_T$ and the bilinear map $e$ are computable in deterministic polynomial time in $\lambda$. In this work, we only consider asymmetric (or Type-III) pairing groups where there exists no efficiently computable homomorphism between $G_1$ and $G_2$. In some cases we will use implicit representation of group elements: for a vector $v$ over $\mathbb{Z}_p$, we define $[v]_1 := g_x^v$ for $s \in \{1, 2, T\}$, where exponentiation is carried out component-wise.

2.2 Attribute-based Encryption

Throughout the paper, we will use a KEM-style definition of ABE. However note that it is implied by the corresponding definition in the PKE setting.

Syntax. An attribute-based encryption (ABE) scheme for some class $P$ consists of four algorithms:

Setup$(1^\lambda, P) \rightarrow (\text{mpk}, \text{msk})$. The setup algorithm gets as input the security parameter $1^\lambda$ and class description $P$. It outputs the master public key $\text{mpk}$ and the master secret key $\text{msk}$. We assume $\text{mpk}$ defines the key space $K$.

Enc$(\text{mpk}, x) \rightarrow (\text{ct}_x, d)$. The encryption algorithm gets as input $\text{mpk}$ and an input $x$. It outputs a ciphertext $\text{ct}_x$ and an encapsulated key $d \in K$.

KeyGen$(\text{mpk}, \text{msk}, y) \rightarrow \text{sk}_y$. The key generation algorithm gets as input $\text{mpk}$, $\text{msk}$ and $y \in P$. It outputs a secret key $\text{sk}_y$.

Dec$(\text{mpk}, x, y, \text{ct}_x, \text{sk}_y) \rightarrow m$. The decryption algorithm gets as input $\text{sk}_y$ and $\text{ct}_x$ such that $P(x, y) = 1$ along with $\text{mpk}$. It outputs a key $d$. It outputs a key $d$. 


We consider PES-ABE, which is a standard ABE scheme augmented with 3 deterministic algorithms

Many-Ciphertext CPA Security. We define security by a game between a challenger and an adversary $A$. The challenger picks a random challenge bit $\beta$ and provides the following oracles to $A$.

- Setup oracle $O_{\text{setup}}$: This oracle can only be queried once and it must be the first query. The challenger runs $\text{Setup}$ to obtain $(mpk, msk)$ and outputs $mpk$ to $A$.

- Ciphertext (or challenge) oracle $O_{\text{ct}}$: On the $i$-th query, $A$ provides $x_i \in \mathcal{X}$. The challenger runs $(ct_i, d_i^{(0)}) \leftarrow \text{Enc}(mpk, x_i)$, chooses a random key $d_i^{(1)} \leftarrow \mathcal{K}$ and outputs $(ct_i, d_i^{(\beta)})$.

- Secret key oracle $O_{\text{sk}}$: On the $j$-th query, $A$ provides $y_j \in \mathcal{Y}$. The challenger runs $sk_j \leftarrow \text{KeyGen}(msk, y_j)$ and outputs $sk_j$.

$O_{\text{ct}}$ and $O_{\text{sk}}$ can be queried adaptively and an arbitrary polynomial number of times. Finally, $A$ outputs a bit $\beta'$. We say that $A$ wins the game if $\beta = \beta'$ and $P(x_i, y_j) = 0$ for all queries $x_i$ and $y_j$.

**Definition 1.** An ABE scheme is adaptively many-ciphertext secure if for all efficient $A$,

$$
\text{Adv}_{\text{ABE}, \mathcal{A}}(\lambda) := |\text{Pr}[\beta = \beta'] - \frac{1}{2}|
$$

is negligible in $\lambda$.

Boolean formulae and MSP Boolean formulae are a common way to model access control. A (monotone) boolean formula consists of and and or gates, where each input is associated with an attribute in the universe of attributes denoted by $U$. Monotone means that an authorized user who acquires more attributes will not lose any privileges. Let $S \subseteq U$ be a set of attributes. We say that $S$ satisfies a boolean formula if we set all inputs of the formula that map to an attribute in $S$ to true and the others to false and the formula evaluates to true.

Monotone span programs (MSP) are a more general class of functions and include boolean formulae. We encode an access structure by a policy $(\mathbf{M}, \pi)$, where $\mathbf{M} \in \mathbb{Z}_p^{n_1 \times n_2}$ and $\pi : [n_1] \rightarrow U$. Note that we can compute $(\mathbf{M}, \pi)$ for any (monotone) boolean formula in polynomial time [39]. Then every row $\mathbf{M}_i$ corresponds to an input to the formula and the number of columns is the same as the number of and gates. If the mapping $\pi$ is not injective, we use the notation $\rho(i) := \{|z | \pi(z) = \pi(i), z \leq i\}$ to denote the $\rho(i)$-th occurrence of attribute $\pi(i)$.

Let $S \subseteq U$ be a set of attributes and $I = \{i | i \in [n_1], \pi(i) \in S\}$ be the indices of rows in $\mathbf{M}$ that are associated with $S$. We say that $(\mathbf{M}, \pi)$ accepts $S$ if the vector $(1, 0, \ldots, 0)$ lies in the span of rows associated with $S$. This means, there exist constants $\gamma_i \in \mathbb{Z}_p$ for $i \in I$ such that $\sum_{i \in I} \gamma_i \mathbf{M}_i = (1, 0, \ldots, 0)$. These constants can be computed in time polynomial in the size of $\mathbf{M}$. On the contrary, $(\mathbf{M}, \pi)$ does not accept $S$ if there exist a vector $\mathbf{w} \in \mathbb{Z}_p^{n_2}$ such that $\mathbf{w}$ is orthogonal to all rows $\mathbf{M}_i$ for $\pi(i) \in S$, but not to $(1, 0, \ldots, 0)$. That means $\langle \mathbf{w}, \mathbf{M}_i \rangle = 0$. W.l.o.g. we can set $\mathbf{w}[1] = 1$.

**Polynomials.** Let $p$ be a prime and $n \in \mathbb{N}$. We denote the set of multi-variate polynomials over $\mathbb{Z}_p$ with indeterminates $\bar{x}_1, \ldots, \bar{x}_n$ by $\mathbb{Z}_p[\bar{x}_1, \ldots, \bar{x}_n]$.

## 3 PES-ABE

We consider PES-ABE, which is a standard ABE scheme augmented with 3 deterministic algorithms $\text{Setup}_0$, $\text{Enc}_0$, $\text{KeyGen}_0$ used in $\text{Setup}$, $\text{Enc}$, $\text{KeyGen}$, $\text{Dec}$ respectively, where:

- $\text{Setup}_0(\lambda, \mathcal{X}, \mathcal{Y})$ outputs $n \in \mathbb{N}$,
- $\text{Enc}_0(x)$ outputs linear functions $c^1 : \mathbb{Z}_p^{m_1} \rightarrow \mathbb{Z}_p^{m_2}$,
- $\text{KeyGen}_0(y)$ outputs linear functions $k^1 : \mathbb{Z}_p^{1+m+mn} \rightarrow \mathbb{Z}_p^{m_1}$, $k^2 : \mathbb{Z}_p^{m} \rightarrow \mathbb{Z}_p^{m_2}$,
and

- **Setup**: Run $\mathcal{G} := (p, G_1, G_2, G_T, c, g_1, g_2) \leftarrow \text{GroupGen}(1^k)$, $n \leftarrow \text{Setup}_0$. Pick $\alpha \leftarrow \mathbb{Z}_p$ and a hash function $\mathcal{H} : [n] \rightarrow G_1$.

Using $\mathcal{H}$, we implicitly define $b \in \mathbb{Z}_p^n$ via $[b[i]]_1 = \mathcal{H}(i)$.

- **Enc**: Run $(c^1, c^2) \leftarrow \text{Enc}_0(x)$. Pick $s \leftarrow \mathbb{Z}_p^w$. Compute $[c^1]_1 := c^1([s \otimes b]_1), [c^2]_2 := c^2([s]_2)$ where $c^2[1] = s[1]$. Output $ct := ([c^1]_1, [c^2]_2_2)$, $\text{kem} := [\alpha s[1]]_T$.

- **KeyGen**: Run $(k^1, k^2) \leftarrow \text{KeyGen}_0(y)$. Pick $r \leftarrow \mathbb{Z}_p^m$. Compute $[k^1]_1 := k^1([\alpha]_1, [r]_1, [b \otimes r]_1), [k^2]_2 := k^2([r]_2)$. Output $sk := ([k^1]_1, [k^2]_2)$.

Note that $\text{Enc}$ and $\text{KeyGen}$ compute the linear functions $c^1, k^1$ “in the exponent” since it only knows $[b]_1$ and not $b$. We also require that $c^1, k^1$ depend only on a polynomial number of entries in $b$, so that $\text{Enc}, \text{KeyGen}$ only need to make a polynomial number of calls to $\mathcal{H}$ to compute $[c^1(s \otimes b)]_1$ and $[k^1(\alpha, r, b \otimes r)]_1$ respectively. Depending on the application, some of these calls to $\mathcal{H}$ can also be precomputed.

**Remark 1 (Decryption).** Note that we can augment PES-ABE with an additional deterministic algorithm $\text{Dec}_0$ used in $\text{Dec}$ where

- $\text{Dec}_0(x, y)$ outputs $e \in \mathbb{Z}_p^{w_1 m_2}, e' \in \mathbb{Z}_p^{w_2 m_2}$;
- $\text{Dec}(\text{mpk}, x, y, ct) = ([c^1]_1, [c^2]_2), sk = ([k^1]_1, [k^2]_2)$: Run $(e, e') \leftarrow \text{Dec}_0(x, y)$. Compute $[k^1 \otimes c^2]_T, [c^1 \otimes k^2]_T$ using $e$, and output $([k^1 \otimes c^2]_T e^T + (c^1 \otimes k^2) e'^T)_{1T}$.

It would then follow from ABE correctness that if $P(x, y) = 1$, $(k^1 \otimes c^2) e^T + (c^1 \otimes k^2) e'^T = \alpha s[1]$. We omit $\text{Dec}_0$ in our presentation and instead, specify and analyze $\text{Dec}$ for correctness directly. This does not affect our security notions and proofs which only refer to $\text{Enc}, \text{KeyGen}, \text{Enc}_0, \text{KeyGen}_0$.

## 4 Symbolic Security of PES-ABE

Following previous work [34], we define symbolic security for PES-ABE, where we replace the inputs $(\alpha, b, s, r) \leftarrow \mathbb{Z}_p \times \mathbb{Z}_p^n \times \mathbb{Z}_p^w \times \mathbb{Z}_p^m$ to the linear functions $(c^1, c^2, k^1, k^2)$ with vectors of formal variables

$$(\tilde{\alpha}, \tilde{b}, \tilde{s}, \tilde{r}) \leftarrow \text{Var} \times \text{Var}^n \times \text{Var}^w \times \text{Var}^m$$

In particular, $c^1(\tilde{s} \otimes \tilde{b}), c^2(\tilde{s}), k^1(\tilde{b} \otimes \tilde{r}), k^2(\tilde{r})$ are now (vectors of) polynomials in $\mathbb{Z}_p[\tilde{\alpha}, \tilde{b}, \tilde{s}, \tilde{r}]$.

### 4.1 Definitions

Fix $x \in X, y \in Y$. ABE correctness tells us that if $P(x, y) = 1$, then

$$\tilde{\alpha} \tilde{s}[1] \in \text{span}(c^1(\tilde{s} \otimes \tilde{b}) \otimes k^2(\tilde{r})\|k^1(\tilde{\alpha}, \tilde{b} \otimes \tilde{r}) \otimes c^2(\tilde{s}))$$

On the other hand, if $P(x, y) = 0$, it should be the case that

$$\tilde{\alpha} \tilde{s}[1] \notin \text{span}(c^1(\tilde{s} \otimes \tilde{b}) \otimes k^2(\tilde{r})\|k^1(\tilde{\alpha}, \tilde{b} \otimes \tilde{r}) \otimes c^2(\tilde{s}))$$

Our basic formulation of symbolic security stipulates something stronger, where we basically replace $\tilde{\alpha} \tilde{s}[1]$ with $\tilde{\alpha} \otimes c^2(\tilde{s})$ and require $c^2(\tilde{s})[1] = \tilde{s}[1]$. In the special case where $c^2(\tilde{s}) = \tilde{s}[1]$ (as is the case when $w_2 = 1$), these two requirements are equivalent.
Definition 2 ((1, 1) Symbolic Security). For all $x \in X$, $y \in Y$ such that $P(x, y) = 0$ we have

$$\text{span}(\tilde{\alpha} \otimes c^2) \cap \text{span}(c^1 \otimes k^2 \| k^1 \otimes c^2) = \{0\},$$

where

$$(\tilde{\alpha}, \tilde{b}) \leftarrow \text{Var} \times \text{Var}^n$$
$$(c^1, c^2) := (c^1(\tilde{s}, \tilde{b}), c^2(\tilde{s})), \tilde{s} \leftarrow \text{Var}^w, (c^1, c^2) \leftarrow \text{Enc}_0(x)$$
$$(k^1, k^2) := (k^1(\tilde{\alpha}, \tilde{r}, \tilde{b} \otimes \tilde{r}), k^2(\tilde{r})) \leftarrow \text{Var}^m, (k^1, k^2) \leftarrow \text{KeyGen}_0(y).$$

The symbolic property captured by this definition will be required to prove many-ciphertext CPA security of our ABE scheme. To capture the ABE security experiment more closely, we extend the definition such that it also include many secret keys, many ciphertexts as well as the public key. We also consider that in the ABE security experiment the adversary may ask for the same $x$ or $y$ multiple times. In Lemma 4 below, we show that this stronger symbolic property is actually implied by the weaker one above.

Definition 3 (Strong Symbolic Security). For all $Q_x, Q_{sk} \in \mathbb{N}$, $X \in \mathcal{X}^{Q_x}, Y \in \mathcal{Y}^{Q_{sk}}$ such that $P(X[i], Y[j]) = 0$ for all $i \in [Q_x]$, $j \in [Q_{sk}]$, we have

$$\text{span}(\tilde{\alpha} \otimes c^2) \cap \text{span}(\tilde{\alpha} \| (1\|b)c^1_X \| k^1_X), (1\|c^2_X \| k^2_X) = \{0\},$$

where

$$(\tilde{\alpha}, \tilde{b}) \leftarrow \text{Var} \times \text{Var}^n$$
$$(c^1, c^2) := (c^1(\tilde{s}, \tilde{b}), c^2(\tilde{s})), \tilde{s} \leftarrow \text{Var}^w, (c^1, c^2) \leftarrow \text{Enc}_0(X[i], \forall i \in [Q_x]),$$
$$(k^1, k^2) := (k^1(\tilde{\alpha}, \tilde{r}, \tilde{b} \otimes \tilde{r}), k^2(\tilde{r})) \leftarrow \text{Var}^m, (k^1, k^2) \leftarrow \text{KeyGen}_0(Y[j], \forall j \in [Q_{sk}],$$
$$c^1_X := (c^1 \| \cdots \| c^1_{Q_{sk}}), c^2_X := (c^2 \| \cdots \| c^2_{Q_{sk}}),$$
$$k^1_Y := (k^1 \| \cdots \| k^1_{Q_{sk}}), k^2_Y := (k^2 \| \cdots \| k^2_{Q_{sk}}).$$

4.2 Relations

Now we can establish the desired implication in the following lemma.

Lemma 1. If a PES-ABE scheme satisfies (1, 1) symbolic security (Definition 3), then it also satisfies strong symbolic security (Definition 3).

The proof follows the high-level strategy laid out in [3] Theorem 4.1] with two main differences: (i) the proof of Claim 2 where we handle $w_2 > 1$ (see also Remark 2) and (ii) Step 2 where we handle many-ciphertext security.

Proof. Fix a PES-ABE satisfying (1, 1) symbolic security as well as $Q_{sk}, Q_x, X, Y$ satisfying the conditions in Definition 3. We want to show that

$$\text{span}(\tilde{\alpha} \otimes c^2) \cap \text{span}(\tilde{\alpha} \| (1\|b)c^1_X \| k^1_X), (1\|c^2_X \| k^2_X) = \{0\}. \quad (2)$$

The proof proceeds in three steps.

Step 1. First, we show that for all $i \in [Q_x]$,

$$\text{span}(\tilde{\alpha} \otimes c^2) \cap \text{span}(c^1 \otimes k^2 \| k^1 \otimes c^2) = \{0\}.$$  

The proof proceeds by contradiction. Suppose on the contrary that there exist $i^* \in [Q_x]$, $e^* \in \mathbb{Z}_p^{w_2}$, $e_j \in \mathbb{Z}_p^{w_1 w_2}, e'_j \in \mathbb{Z}_p^{w_1 w_2}$ for all $j \in [Q_{sk}]$ such that $e^* \neq 0$ and

$$(\tilde{\alpha} \otimes c^2; e^*) \cdot e^* = \sum_{j \in [Q_{sk}]} (c^1_j \otimes k^2_j) \cdot e^*_j + (k^1_j \otimes c^2_j) \cdot e'^*_j. \quad (3)$$

We claim that $\{e_j, e'_j\}_{j \in [Q_{sk}]}$ then satisfies

10
- Claim 1: \((c_1 \otimes k_2^2) \cdot e_j^T + (k_j^1(0, \tilde{r}_j, \tilde{b} \otimes \tilde{r}_j) \otimes c_2^2) \cdot e_j^T = 0\) for all \(j \in [Q_{ab}]\).
- Claim 2: there exists \(j^* \in [Q_{ab}], \mu \in \mathbb{Z}_p^{u_2} \) such that \(\mu \neq 0\) and \((\tilde{a} \otimes c_2^2) \cdot \mu^T = (k_j^1(0, \tilde{a}, 0, 0) \otimes c_2^2) \cdot e_{j^*}^T\).

Combining the two claims with the fact that \(k_j^1 = k_j^1(0, \tilde{r}_j, \tilde{b} \otimes \tilde{r}_j) + k_j^2(\tilde{a}, 0, 0)\), we have

\[
(\tilde{a} \otimes c_2^2) \cdot \mu^T = (c_1^1 \otimes k_2^2) \cdot e_j^T + (k_j^1 \otimes c_2^2) \cdot e_{j^*}^T,
\]

which contradicts \((1, 1)\) symbolic security since \(P(X[i^*], Y[j^*]) = 0\). It remains to establish Claims 1 and 2 to complete the proof:

- Fix \(j \in [Q_{ab}]\). Claim 1 follows from evaluating (3) on \(\tilde{a} = 0, \tilde{r}_j' = 0\) for all \(j' \in [Q_{ab}] \setminus \{j\}\).
- Next, evaluating (3) on \(\tilde{r}_j = 0\) for all \(j \in [Q_{ab}]\) yields

\[
0 \neq (\tilde{a} \otimes c_2^2) \cdot e_j^T = \sum_{j \in [Q_{ab}]} (k_j^1(\tilde{a}, 0, 0) \otimes c_2^2) \cdot e_{j^*}^T.
\]

Therefore, there exists \(j^* \in Y\) such that \((k_j^1, (\tilde{a}, 0, 0) \otimes c_2^2) \cdot e_{j^*}^T \neq 0\). Moreover, since the polynomial \(k_j^1(\tilde{a}, 0, 0)\) is linear in \(\tilde{a}\), there exists \(\mu \neq 0\) such that \((k_j^1, (\tilde{a}, 0, 0) \otimes c_2^2) \cdot e_{j^*}^T = (\tilde{a} \otimes c_2^2) \cdot \mu^T\) and Claim 2 follows.

**Step 2.** We show that

\[
\text{span}(\tilde{a} \otimes c_2^2) \cap \text{span}(e_1^X \otimes k_2^1 \parallel k_2^1 \otimes c_2^X) = \{0\}.
\]

As in the previous step, the proof proceeds by contradiction. Suppose the above statement is false, which means there exist \(e_i^* \in \mathbb{Z}_p^{u_2}, e_i^* \in \mathbb{Z}_p^{Q_a \cdot m_2}, e_i^* \in \mathbb{Z}_p^{Q_a \cdot m_2} \) for all \(i \in [Q_{ac}]\) and \(i^* \in [Q_{ac}]\) such that

\[
\sum_{i \in [Q_{ac}]} (\tilde{a} \otimes c_i^2) \cdot e_i^{T*} = \sum_{i \in [Q_{ac}]} (c_i^1 \otimes k_i^2) \cdot e_i^T + (k_i^1 \otimes c_i^2) \cdot e_i^{T*},
\]

and \(e_i^{T*} \neq 0\). We evaluate (4) on \(\tilde{s}_i^* = 0\) for all \(i \in [Q_{ac}]\) and get

\[
(\tilde{a} \otimes c_i^2) \cdot e_i^{T*} = (c_i^1 \otimes k_i^2) \cdot e_i^T + (k_i^1 \otimes c_i^2) \cdot e_i^{T*}.
\]

That is, \(\text{span}(\tilde{a} \otimes c_i^2) \cap \text{span}(c_i^1 \otimes k_i^2 || k_i^2 \otimes c_i^X) \neq \{0\}\), which contradicts what we showed in Step 1.

**Step 3.** We now prove (2), which also proceeds by contradiction. Suppose on the contrary that

\[
\text{span}(\tilde{a} \otimes c_2^X) \cap \text{span}(\tilde{a} || (1 \parallel \tilde{b}) || k_2^X \parallel k_2^X) \cap (1 || c_2^X \parallel k_2^X) \neq \{0\}.
\]

Then there exist \(e^* \in \mathbb{Z}_p^{Q_a \cdot u_2}, e_{PK} \in \mathbb{Z}_p^{2^{u_2}}, e_X \in \mathbb{Z}_p^{Q_a \cdot (u_1 + u_2 + u_1 + u_2 + u_2)}, e_Y \in \mathbb{Z}_p^{Q_a \cdot (m_1 + m_2 + m_1 + m_2 + m_2)}, e_{XY} \in \mathbb{Z}_p^{Q_a \cdot (m_1 + m_2 + m_1 + m_2)}\),

\[
(\tilde{a} \otimes c_2^X) \cdot e^{T*} = (1 || \tilde{a} || (1 \parallel \tilde{b}) || k_2^X \parallel k_2^X) \cdot e_{PK} + (c_1^X \parallel c_2^X \parallel k_2^X \parallel \tilde{b} \parallel c_2^X) \cdot e_X^T
\]

+ (k_1^2 || k_2^2 || k_2^X \parallel \tilde{b} \parallel k_2^X) \cdot e_Y^T
\]

+ (c_1^X \parallel k_2^2 \parallel k_2^X \parallel c_2^X) \cdot e_{XY}^T
\]

and \(e^* \neq 0\). First, we look at the first three terms on the RHS of (5):

- Evaluating (5) on \(\tilde{a} = 0, \tilde{r}_Y = 0\), and \(\tilde{s}_X = 0\) yields \((1 || 0 \parallel \tilde{b}) \cdot e_{PK} = 0\).
- Evaluating (5) on \(\tilde{a} = 0, \tilde{r}_Y = 0\) yields \((1 || 0 \parallel \tilde{b}) \cdot e_Y^T = 0\).
- Evaluating (5) on \(\tilde{s}_X = 0\) yields \((1 || \tilde{a} \parallel (1 \parallel \tilde{b}) \cdot e_{PK} + (k_1^2 \parallel k_2^2 \parallel k_2^2 \parallel (k_2^X \parallel k_2^X \parallel b \parallel k_2^X) \cdot e_Y^T = 0\).

Subtracting the first equality from the sum of the second and third implies that the sum of the first three terms on the RHS of (5) is 0. This means

\[
(\tilde{a} \otimes c_2^X) \cdot e^{T*} = (c_1^X \parallel k_2^X || k_2^X \parallel c_2^X) \cdot e_{XY}^T
\]

which contradicts what we showed in Step 2.
Lemma 2 (Schwartz-Zippel Lemma). We first recall a useful lemma that is commonly used to prove security in the GGM.

Remark 2 (handling \( w_1 > 1 \)). In the proof of the analogue of Claim 2 in \( [9] \), they start with

\[
\tilde{a} c_2^i[1] = \sum_{j \in [Q_w]} (k_j^i (\tilde{a}, 0, 0) \otimes c_j^2) \cdot e_j^2.
\]

They show that if \( w_2 = 1 \) (a requirement mentioned in the proof\( [7] \) but not in the theorem statement), then there exists \( j^* \in Y \) such \( \mu \cdot c_2^i[1] = k_j^i (1, 0, 0) \otimes c_j^2 \cdot e_j^2 \) and \( \mu \neq 0 \). However, if we allow \( w_2 > 1 \), then this claim does not hold in general. In particular, it could be that for all \( j \), \( c_2^i[1] \) only appears in a linear combination with other elements of \( c_2^i \), which then all together sum up to \( c_2^i[1] \). For this reason, we need to strengthen our definition accordingly.

5 Optimal ABE Security in the GGM

We prove symbolic security of PES-ABE implies optimal, adaptive security in the generic group model (GGM). For that, we first recall the generic group model.

5.1 Generic Group Model

In the generic group model, an adversary can perform group operations only via oracle access. We adopt the model by Maurer \( [43] \) extended to the pairing group setting, where apart from the group operation, the adversary can also compute the pairing via an oracle. A third party implements the pairing group model the hash function in our scheme as random oracle, so we additionally provide an oracle \( O \).

In game \( G_0 \) in Figure 2, we model the ABE security game from Section 2.2 in the GGM. That is, the adversary also gets access to oracles \( \mathcal{O}_{mpk} \), \( \mathcal{O}_{ct} \) and \( \mathcal{O}_{sk} \). On each query, the corresponding oracle returns the current length of all modified lists from which the adversary can deduce the corresponding handles since length of ciphertexts and secret keys follow from the definition of the scheme. Furthermore, we model the hash function in our scheme as random oracle, so we additionally provide an oracle \( H \), which also modify the lists. The adversary can then use these indices in further group operation and equality queries as described above.

5.2 Security

The following theorem states that symbolic security implies optimal, adaptive security in GGM.

Theorem 1. Let \( \lambda \in \mathbb{N} \) be the security parameter and \( A \) be an adversary that on input \( (1^\lambda, p) \) makes \( Q_{\text{add}}, Q_{\text{pair}}, Q_{\text{ct}}, Q_{\text{sk}}, Q_{\text{eq}} \) queries to oracles \( \mathcal{O}_{\text{add}}, \mathcal{O}_{\text{pair}}, \mathcal{O}_{\text{ct}}, \mathcal{O}_{\text{sk}}, \mathcal{O}_{\text{eq}} \) and \( Q_H \) queries to the random oracle \( H \). If PES-ABE is \( (1,1) \) symbolically secure (Definition \( [3] \)), then it is adaptively Many-CT secure in the GGM. In particular,

\[
\text{Adv}_{\text{ABE},A}^{\text{GM}}(\lambda) \leq 3 \cdot (Q_H + (w' + 1) \cdot Q_{\text{ct}} + m' \cdot Q_{\text{sk}} + Q_{\text{add}} + Q_{\text{pair}})^2 / p,
\]

where \( w' := w_1 + w_2 \) and \( m' := m_1 + m_2 \).

We first recall a useful lemma that is commonly used to prove security in the GGM.

Lemma 2 (Schwartz-Zippel Lemma). For any prime \( p \) and \( t \in \mathbb{N}^* \), any polynomial \( f \in \mathbb{Z}_p[\bar{x}_1, \ldots, \bar{x}_t] \) of degree \( d > 0 \),

\[
\Pr[f(\bar{x}) = 0] \leq d / p,
\]

where the probability is taken over \( \bar{x} \in \mathbb{Z}_p^t \).

\(^{9}\) On page 662, they wrote "since we assumed \( w_1 = 0 \). Here, \( c^2(\bar{s}) \) corresponds to \( \bar{S} = (S_0, \ldots, S_{w_1}) \) in \( [9] \)."
Proof. The proof will be done by a hybrid argument over the queries to oracle $\mathcal{O}_{\text{eq}}$ in lists $A$ and $G$.

We use lists $L_1$, $L_2$ and $L_T$ to store scalars in $\mathbb{Z}_p$ which correspond to public parameters, ciphertexts and secret keys in $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_T$, respectively. To keep track of queries to $\mathcal{O}_\text{ct}$ and $\mathcal{O}_\text{sk}$, we store them in lists $X$ and $Y$.

We also already store polynomials in separate lists which we denote by $L_1^\perp$, $L_2^\perp$ and $L_T^\perp$. As in the previous section, we label formal variables with a tilde, e.g. $\tilde{a}$ is the formal variable corresponding to the master secret key $\alpha \in \mathbb{Z}_p$. Note that in $\mathcal{O}_{\text{mpk}}$ we already define formal variables $\tilde{b}_i$ for all $i \in \mathcal{U}$, but they

\begin{figure}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Games $\{\mathcal{G}_\mu\}_{\mu \in [0,Q_{\text{eq}}]}$ & Games $\{\mathcal{G}_\mu\}_{\mu \in [0,Q_{\text{eq}}]}$ \\
\hline
\hline
\begin{itemize}
\item $i = j := 0$, $\nu \leftarrow 0$, $X = Y = \mathcal{H} := \emptyset$
\item for $s \in \{1,2,T\}$: $L_s := \emptyset$, $L_s^\perp := \emptyset$
\item $\beta \leftarrow \{0,1\}$
\item $\beta' \leftarrow \mathcal{A}_{\text{mpk}}(\mathcal{O}_{\text{mpk}},\mathcal{O}_{\text{ct}},\mathcal{O}_{\text{sk}},\mathcal{H}(\lambda,p))$
\item return $[\beta = \beta']$ and
\item $[P(X[i],Y[j]) = 0 \forall i \in [Q_{\text{eq}}], j \in [Q_{\text{sk}}]]$
\end{itemize}
\end{tabular}
\begin{tabular}{|c|c|}
\hline
$\mathcal{O}_{\text{eq}}(x \in X)$ & $\mathcal{O}_{\text{eq}}(x \in X)$ \\
\hline
\begin{itemize}
\item $c_1^j(\tilde{a} \otimes \tilde{b}),c_2^j(\tilde{a})) \leftarrow \text{Enc}_0(x)$
\item $a_1, \tilde{a} \leftarrow \mathcal{Z}_p$, $\tilde{a} \leftarrow \mathcal{V}_n$
\item $d_1^{(0)} := \text{os}_1(1), d_1^{(1)} := \omega_1, \neq, \mathcal{Z}_p$
\item $d_1^{(0)} := \tilde{a}s_i(1), d_1^{(1)} := \omega_1 \neq \mathcal{V}_n$
\item $L_1.$ append$(c_1^j(\tilde{s} \otimes \tilde{b})), L_2.$ append$(c_2^j(\tilde{s})), L_T.$ append$(d_1^{(0)})$
\item $L_1.$ append$(c_1^j(\tilde{s} \otimes \tilde{b})), L_2.$ append$(c_2^j(\tilde{s})), L_T.$ append$(d_1^{(1)})$
\item $X.$ append$(x)$, $\nu := \nu + 1$
\item return $[L_1, |L_2|, |L_T|$
\end{itemize}
\end{tabular}
\caption{Games $\mathcal{G}_\mu$ for $\mu \in [0,Q_{\text{eq}}]$ for the proof of Theorem 1. Note that the games only differ in oracle $\mathcal{O}_{\text{eq}}$ (which depends on $\mu$). Here, $\mathcal{G}_0$ corresponds to the GGM experiment that makes only use of components in light gray frames, whereas $\mathcal{G}_{Q_{\text{eq}}}$ makes only use of components in dark gray frames. W.l.o.g. we assume that no query to $\mathcal{O}_{\text{add}}, \mathcal{O}_{\text{pair}}, \mathcal{O}_{\text{eq}}$ contains indices $i',j' \in \mathbb{N}$ which exceed the size of the involved lists.}
\end{figure}

We will now state the full proof of Theorem 1. In fact, it is similar to that in [6] Theorem 3.3. The latter only considers single-ciphertext ABE security, and achieves an additional loss of $Q_{\text{sk}}$, since they apply Schwartz-Zippel to polynomials of degree $Q_{\text{sk}}$ in order to handle rational fractions arising in their schemes.

\textbf{Proof.} The proof will be done by a hybrid argument over the queries to oracle $\mathcal{O}_{\text{eq}}$. The first hybrid $\mathcal{G}_0$ is the many-CT CPA security game for the ABE scheme in the generic group model. We will proceed in two main steps.

1. In each hybrid, we replace the check whether two elements are equal by checking whether their corresponding polynomials are equal;
2. in the last game $\mathcal{G}_{Q_{\text{eq}}}$, we use the symbolic security property of the ABE scheme to show that outputs are independent of the challenge bit $\beta$.

Now consider the games $\mathcal{G}_0$-$\mathcal{G}_{Q_{\text{eq}}}$ given in Figure 2. In the following we denote the advantage of adversary $\mathcal{A}$ in $\mathcal{G}_\mu$ by $\text{Adv}_{\mu}$.

\begin{itemize}
\item Game $\mathcal{G}_0$. This game captures the ABE security experiment for PES-ABE as described in Section 3 in the generic group model, thus
\end{itemize}

$$\text{Adv}_{\mu} = \text{Adv}_{\mathcal{G}_{\text{ABE}},\mathcal{A}(\lambda)}.$$
are only added to $L_1$, when the random oracle is queried on $i$. All queries to $H$ are additionally stored in a set $\mathcal{H}$.

**Game $\{G_{\mu}\}_{\mu \in \mathbb{Q}_n}$.** We now proceed using a hybrid argument. Note that the only difference between $G_{\mu-1}$ and $G_{\mu}$ lies in how we answer the $\mu$-th query to oracle $\mathcal{O}_{eq}$: in $G_{\mu-1}$, we are using $L_T$ thus comparing scalars, and in $G_{\mu}$, we are using $L_T^\sim$ and thus comparing polynomials. Let $(s, i, j)$ denote the $\mu$-th query to oracle $\mathcal{O}_{eq}$. We claim that

$$|\text{Adv}_{\mu-1} - \text{Adv}_{\mu}| \leq \Pr[(L_s[i'] \neq L_s[j']) \land (L_s^\sim[i'] = L_s^\sim[j'])] \leq \frac{3}{p},$$

where the randomness is over $\alpha \overset{\$}{\leftarrow} \mathbb{Z}_p, b \overset{\$}{\leftarrow} \mathbb{Z}_p^n, s_i \overset{\$}{\leftarrow} \mathbb{Z}_p^w, r_j \overset{\$}{\leftarrow} \mathbb{Z}_p^m, \omega_i \overset{\$}{\leftarrow} \mathbb{Z}_p$. Let $d_{X}^{(j)} := (d_1^{(j)}|| \cdots || d_{X}^{(j)})$ and define $c_X^1, c_X^2, k_1^\sim, k_2^\sim$ as in Definition 2.

It is easy to see that:

$$L_T^\sim \subseteq \text{span}(c_{mpk} \parallel H \parallel c_{X} \parallel k_1^\sim)$$
$$L_2^\sim \subseteq \text{span}(c_{mpk} \parallel c_X \parallel k_2^\sim)$$
$$L_T^\sim \subseteq \text{span}(d_X^{(j)} \parallel \text{span}(c_{mpk} \parallel (1||b||c_X^1||k_1^\sim) \otimes (1||c_X^2||k_2^\sim)) \parallel \text{span}(c_{mpk} \parallel \text{span}(c_{X} \parallel c_X \parallel k_2^\sim)))$$

(6)

where span$(\cdot)$ captures queries to $\mathcal{O}_{eq}$. Since $c_X^1, k_1^\sim$ have degrees at most 2 and $c_X^2, k_2^\sim$ have degrees at most 1, the polynomials in $L_T^\sim, L_T^\sim$ and $L_T^\sim$ have degrees at most 1, 2 and 3 respectively (in the formal variables $\bar{a}, b, \bar{s}_X, \bar{Y}, \bar{X}$). The claim then follows readily from the Schwartz-Zippel Lemma.

**Game $G_{Q_n}$.** We show that $\text{Adv}_{Q_n} = 0$. In particular, we claim that the view of the adversary in $G_{Q_n}$ is independent of the challenge bit $\beta$. Observe that the only leakage about $\beta$ is the output of $\mathcal{O}_{eq}$ on a query of the form $(T, i', j')$. By [5], there exists $e_i', e_{j'} \in \mathbb{Z}_p$ and $v_i', v_{j'} \in \text{span}(\alpha||1||b||c_X^1||k_1^\sim) \otimes (1||c_X^2||k_2^\sim))$ such that

$$L_T^\sim[i'] = \bar{d}_X^{(j)} \cdot e_i' + v_i', \quad L_T^\sim[j'] = \bar{d}_X^{(j)} \cdot e_{j'} + v_{j'}.$$

It suffices to show that

$$L_T^\sim[i'] = L_T^\sim[j'] \iff e_i' = e_{j'} \land v_i' = v_{j'}$$

since the RHS is independent of $\beta$.

First, we show that for both cases $\beta = 0$ and $\beta = 1$, it holds that

$$\text{span}(\bar{d}_X^{(j)}) \cap \text{span}(\alpha||1||b||c_X^1||k_1^\sim) \otimes (1||c_X^2||k_2^\sim)) = \{0\},$$

(7)

where $\bar{d}_X^{(j)} = (\bar{a}s_i[1]|\cdots||\bar{a}s_{Q_n}[1])$ and $\bar{d}_X^{(1)} = (\bar{Q}_1||\cdots||\bar{Q}_{Q_n})$.

- We first look at the case $\beta = 1$. Obviously, (7) holds since $\bar{Q}_1$ are fresh random variables for all challenges and do not appear anywhere else.

- For $\beta = 0$ first note that since $c_X^2[1] = s_i[1]$, we have

$$\text{span}(\bar{d}_X^{(0)}) \subseteq \text{span}(\alpha \otimes c_X^1) \Rightarrow \text{span}(\bar{d}_X^{(0)}) \subseteq \text{span}(\alpha \otimes c_X^1).$$

Hence, (7) holds due to strong symbolic security of the ABE scheme, which in turn follows from (1, 1) symbolic security via Lemma 1.

From (7), we have

$$L_T^\sim[i'] = L_T^\sim[j'] \iff \bar{d}_X^{(j)} \cdot e_i' = \bar{d}_X^{(j)} \cdot e_{j'} \land v_i' = v_{j'}.$$

It remains to argue that for both $\beta = 0$ and $\beta = 1$

$$\bar{d}_X^{(j)} \cdot e_i' = \bar{d}_X^{(j)} \cdot e_{j'} \iff e_i' = e_{j'}.$$
This follows from the fact that all terms in $\tilde{d}_X^{(i)}$ are mutually independent. For this, note that $\tilde{s}_i[1]$ as well as $\tilde{\omega}_i$ are mutually independent for all $i$ and we get $Adv_{Q_{eq}} = 0$.

Summing everything up, we obtain

$$Adv_{GGM}(\lambda) \leq \frac{3 \cdot Q_{eq}}{p}.$$ 

Then, observing that

$$Q_{eq} \leq (|L_1| + |L_2| + |L_T|)^2$$

$\leq (Q + (w_1 + w_2 + 1) \cdot Q_{ct} + (m_1 + m_2) \cdot Q_{sk} + Q_{add} + Q_{pair})^2$

yields the bound stated in the theorem.

Note that the upper bound on $Q_{eq}$ comes from the number of elements the adversary observes during the GGM experiment. Thus, $Q_{eq} \leq (|L_1| + |L_2| + |L_T|)^2$ which corresponds to the fact that a real-world adversary can detect collisions amongst the group elements it has seen so far in time quasi-linear in $|L_1| + |L_2| + |L_T|$ via sorting, but needs to make $(|L_1| + |L_2| + |L_T|)^2$ queries to $Q_{eq}$ to detect collisions.

6 Our Schemes: Putting Everything Together

We now show that the FABEO CP-ABE and KP-ABE schemes for monotone span programs described in Figure 1 satisfy the PES-ABE framework and (1, 1) symbolic security described in Section 2.2. Combined with the statements from Lemma 1 and Theorem 1, this establishes optimal, adaptive security of our CP-ABE and KP-ABE schemes in GGM (Corollaries 1 and 2). Nonetheless, we additionally prove selective security of our CP-ABE and KP-ABE schemes under the more conservative q-type assumptions in Appendix 3.

6.1 CP-ABE

Our CP-ABE scheme is shown in Figure 1. It builds upon the pair encoding scheme 11 in 7 and that in Appendix B.1 in 3 and extends them by attribute hashing and multi-use of attributes. In particular, we can describe the underlying PES-ABE as follows.

- $\text{Setup}_0$. Output $n := |U| + 1$.
- $\text{Enc}_0(M, \pi)$. Set $w = n_1 + \tau$, $w_1 = n_1$, $w_2 = \tau + 1$, and output $(c_1, c_2)$ where we parse $s$ as $s \parallel v \parallel s'$ and

  $$c_1(s \otimes b) := (M_i(s_i \parallel v)^\top \cdot b[|U| + 1] + s'[\rho(i)] \cdot b[\pi(i)])_{i \in [n_1]},$$

  $$c_2(s_1) := (s_1 || s').$$

- $\text{KeyGen}_0(S)$. Set $m = 1$, $m_1 = |S| + 1$, $m_2 = 1$, and output $(k_1, k_2)$ where we parse $r$ as $(r)$ and

  $$k_1(\alpha, r, b \otimes r) := (\alpha + rb[|U| + 1] || (rb[u])_{u \in S}),$$

  $$k_2(r) := (r).$$

$\text{Correctness}$. Let $ct = (ct_1, (ct_{2,i})_{i \in [\tau]}, (ct_{3,i})_{i \in [n_1]})$ be a ciphertext for $(M, \pi)$ and $sk = (sk_1, (sk_{2,u})_{u \in S}, sk_3)$ be a secret key for $S$ as defined in Figure 1. Further let $b[u]$ such that $H(u) = b[u]$ and $b'$ such that $H(|U| + 1) = b'[u]$. If $S$ satisfies $(M, \pi)$, then there exist constants $(\gamma_i)_{i \in [n_1]}$ such that $\sum_{i \in I} \gamma_i M_i = (1, 0, \ldots, 0)$ and decryption computes

1. $e(g_1^\alpha, H(|U| + 1)^{\gamma_i}) = [\alpha s_1 + b' r s_1]_T$

2. $\prod_{i \in [\tau]} e(\prod_{i \rho(i) = j} H(\pi(i))^{\gamma_i r}, g_2^{\gamma_{[j]}}) = [r \sum_{j \in [\tau]} \sum_{i \in I, \rho(i) = j} \gamma_i b[\pi(i)] s'[j]]_T$

3. $e(\prod_{i \in I} H(|U| + 1)^{\gamma_i M_i(s_1 \parallel v)^\top \cdot H(\pi(i))^{\gamma_i s'[\rho(i)]}}, g_2^\gamma) = [br' \sum_{i \in I} \gamma_i M_i((s_1 || v)^\top)_T \cdot [r \sum_{i \in I} \gamma_i b[\pi(i)] s'[\rho(i)]]_T] = (2)\gamma = (2)$

Note that by definition of $\rho$, (2) and the second term of (3) are the same. Thus computing $(1) \cdot (2)/(3)$ yields $d = [\alpha s_1]_T.$
Symbolic Security. We need to show that for all \((M, \pi) \in \mathcal{X}, S \in \mathcal{Y}\) such that \(P((M, \pi), S) = 0\), it holds that

\[
\text{span}(\alpha \otimes (s_1 || s')) \cap \text{span}((M_i(s_1, v)^T \bar{y}' + s'[\rho(i)]b[i])_{i \in [n_1]} \otimes \bar{r}((\alpha + \bar{r}'(\bar{r}b[u])_{u \in S}) \otimes (s_1 || s')) = \{0\},
\]

where we define \(\bar{r}' \defeq \bar{r}[u] + 1\).

We prove this property by contradiction. So assume there exists \(e^* \in \mathbb{Z}_p^2, e, e^{(1)}, e^{(2)}, e^{(3)}\) such that \(e^* \neq 0\) and

\[
(\alpha \otimes (s_1 || s')) \cdot e^{*T} = (M_i(s_1, v)^T \bar{y}' + s'[\rho(i)]b[i])_{i \in [n_1]} \cdot e^{T} + ((\alpha + \bar{r}'(\bar{r}b[u]) \otimes (s_1 || s')) \cdot e^{(1)}\cdot e^{(2)} + ((\alpha + \bar{r}'(\bar{r}b[u]) \otimes (s_1 || s')) \cdot e^{(3)}\cdot e^{(3)}).
\]

Now we use the fact that \(P((M, \pi), S) = 0\). Recall that this means that there exists a vector \(w \in \mathbb{Z}_p^n\) such that \(\langle w, M_i \rangle = 0\) for all \(i \in \mathcal{S}\) and that \(w[1] = 1\). So evaluating on \((s_1 || v) = w\) gives us

\[
(\alpha \otimes (1 || s')) \cdot e^{*T} = (s'[\rho(i)]b[i])_{i \in [n_1]} \cdot e^{T} + (M_iw^T \bar{y}' + s'[\rho(i)]b[i])_{i \in [n_1]} \cdot e^{T} + ((\alpha + \bar{r}'(\bar{r}b[u]) \otimes (1 || s')) \cdot e^{(1)} + ((\alpha + \bar{r}'(\bar{r}b[u]) \otimes (1 || s')) \cdot e^{(2)} + ((\alpha + \bar{r}'(\bar{r}b[u]) \otimes (1 || s')) \cdot e^{(3)}\cdot e^{(3)}),
\]

where we split \(e\) into two vectors \(\tilde{e} \in \mathbb{Z}_p^{[S]}\) and \(e' \in \mathbb{Z}_p^{[S]\text{-cs}}\), capturing those rows of \(M\) that belong to \(u \in S\) and those that do not belong to an attribute in \(S\). Note that the monomials \([\bar{r}b[u] \otimes s']_{u \in S}\) only appear in the first and the last term. By definition of \(\rho\), the monomials \(s'[\rho(i)]b[i])_{i \in [n_1]} \cdot e^{T} + \alpha\cdot e^{(2)} + \alpha\cdot e^{(3)}\) must be 0. Further looking at monomials on the RHS, \(s'[\rho(i)]b[i])_{i \in [n_1]} \cdot e^{T}\) for \(i \notin S\) and \(\bar{r}b[u]\) for \(u \in S\) are also mutually distinct and only appear in one of the terms, thus \(e\) as well as \(e^{(2)}\) must be 0. Therefore, the following equation must be satisfied

\[
(\alpha \otimes (1 || s')) \cdot e^{*T} = ((\alpha + \bar{r}'(\bar{r}b[u]) \otimes (1 || s')) \cdot e^{(3)}\cdot e^{(3)}),
\]

which leads to a contradiction that \(e^* \neq 0\) since \(\bar{r}'(\bar{r}b[u])\) only appears on the RHS.

Corollary 1. Let \(\lambda \in \mathbb{N}\) be the security parameter and \(A\) be an adversary that on input \((1^k, p)\) makes \(Q_{\text{op}}\) group operation queries to oracles \(O_{\text{add}}\) and \(O_{\text{pair}}\), as well as \(Q_{\text{ct}}, Q_{\text{sk}}\) queries to oracles \(O_{\text{ct}}, O_{\text{sk}},\) and \(Q_{\text{H}}\) queries to the random oracle \(H\). \(\text{CP-ABE}\) is adaptively secure in the GGM such that

\[
\text{Adv}_{\text{GGM}}^{\text{CP-ABE}}(A, (\lambda) \leq \frac{3 \cdot (Q_{\text{H}} + (n_1 + 3) \cdot Q_{\text{ct}} + (|S| + 2) \cdot Q_{\text{sk}} + Q_{\text{op}}) + \frac{2}{p}}{
\}
\]

where \(|S|\) is the maximum size of the attribute sets queried to \(O_{\text{sk}}\) and \(n_1\) is the maximum number of rows of \(M\) queried to \(O_{\text{ct}}\).

6.2 KP-ABE

Our KP-ABE scheme is shown in Figure 1. It builds upon the pair encoding scheme 9 in [7] and that in Appendix B.1 in [3] and extends them by attribute hashing and multi-use of attributes. The underlying PES-ABE can be described as follows.

- **Setup**. Output \(\kappa \defeq |U|\).
- **Enc**. Set \(w = 1, w_1 = |S|, w_2 = 1\), and output \((c^1, c^2)\) where we parse \(s\) as \((s)\) and

\[
c^1(s \otimes b) := (sb[u])_{u \in S}, \quad c^2(s_1) := (s).
\]

- **KeyGen**. Set \(m = r + n_2 - 1, m_1 = n_1, m_2 = r\), and output \((k^1, k^2)\) where we parse \(r\) as \((r'||v)\), and and output

\[
k^1(\alpha, r, b \otimes r) := (M_i(\alpha || v)^T + r'[\rho(i)]b[i])_{i \in [n_1]} \quad k^2(r) := (r')
\]

16
Correctness. Let \( ct = ((ct_{1,u})_{u \in S}, ct_{2}) \) be a ciphertext for \( S \) and \( sk = ((sk_{1,j})_{j \in [\tau]}, (sk_{2,i})_{i \in [n_1]}) \) be a secret key for \((M, \pi)\) as defined in Figure 1. Further let \( b[u] \) such that \( H(u) = g_1^{b[i]} \). If \( S \) satisfies \((M, \pi)\), then there exist constants \((\gamma_i)_{i \in [n_1]}\) such that \( \sum_{i \in I}{\gamma_i M_i} = (1, 0, \ldots, 0) \) and decryption computes

\[
(1) \quad e(\prod_{i \in I}{(g_1^{M_i} (\alpha \| \nu)^T). \cdot H(\pi(i))^{\gamma_i}. g_2^{\tau}}) = [s \sum_{i \in I}{\gamma_i M_i (\alpha \| \nu)^T}]^T. \cdot [s \sum_{i \in I}{\gamma_i b[i \| \tau]^T}]^T.
\]

\[
(2) \quad \prod_{j \in [\tau]}{e(\prod_{i \in I, \rho(i) = j}{H(\pi(i))^{\gamma_s^j}. g_2^{\rho(i)}})} = [s \sum_{j \in [\tau]}{\sum_{i \in I, \rho(i) = j}{\gamma_i b[i \| \tau]^T}}]^T.
\]

Note that by definition of \( \rho \), the second term of (1) is the same as (2). Thus computing \((1)/(2)\) yields \( d = [\alpha s]_T \).

Symbolic Security. Since \( |s| = 1 \), we need to show that for all \( S \in \mathcal{X}, (M, \pi) \in \mathcal{Y} \) such that \( P(S, (M, \pi)) = 0 \), it holds that

\[ \hat{\alpha} s \notin \text{span}((s \tilde{b}[u] \| \tilde{r})_{u \in S} \otimes \tilde{r}|| (M_i (\alpha \| \nu)^T + \tilde{r}[\rho(i) \| \tilde{b}[\pi(i)]), \gamma_i \in [n_1] \otimes \tilde{s}) . \]

We prove this property by contradiction. So assume there exists \( e, e' \) such that

\[ \hat{\alpha} s = (s \tilde{b}[u] \| \tilde{r})_{u \in S} \cdot e^T + (M_i (\alpha \| \nu)^T \tilde{s} + \tilde{r}[\rho(i) \| \tilde{b}[\pi(i)]), \gamma_i \in [n_1] \otimes \tilde{e'}^T . \]

Now we use the fact that \( P(S, (M, \pi)) = 0 \). Recall that this means that there exists a vector \( w \in \mathbb{Z}_p^{n_2} \) such that \( \langle w, M_i \rangle = 0 \) for all \( i \in S \) and that \( w[1] = 1 \). So evaluating on \( (\alpha \| \nu) \) gives us

\[ \hat{\alpha} s = (s \tilde{b}[u] \| \tilde{r})_{u \in S} \cdot e^T + (M_i (\alpha \| \nu)^T \tilde{s} + \tilde{r}[\rho(i) \| \tilde{b}[\pi(i)]), \gamma_i \in [n_1] \otimes \tilde{e'}^T + (M_i (\alpha \| \nu)^T \tilde{s} + \tilde{r}[\rho(i) \| \tilde{b}[\pi(i)]), \gamma_i \in [n_1] \otimes \tilde{e'}^T . \]

where we split \( e' \) into two vectors \( \tilde{e}' \in \mathbb{Z}_p^{n_1 - |S|} \) and \( e' \in \mathbb{Z}_p^{|S|} \), capturing those rows of \( M \) that belong to attributes \( u \in S \) and those that do not belong to an attribute in \( S \). Note that the monomials \( (s \tilde{b}[u] \| \tilde{r})_{u \in S} \) only appear in the first two terms. By definition of \( \rho \), the monomials \( \tilde{r}[\rho(i) \| \tilde{b}[\pi(i)] \tilde{s} \) in the second term are mutually distinct. Thus, we must have \( \tilde{e}'[i] = -e'[j] \) for all \( i \in [n_1] \) such that \( \pi(i) \in S \) and unique indices \( j \), while all other entries in \( e' \) must be 0. Therefore, the following equation must be satisfied as well

\[ \hat{\alpha} s = (M_i (\alpha \| \nu)^T \tilde{s} + \tilde{r}[\rho(i) \| \tilde{b}[\pi(i)]), \gamma_i \in [n_1] \otimes \tilde{e'}^T . \]

However, this leads to a contradiction since the monomials \( (\tilde{r}[\rho(i) \| \tilde{b}[\pi(i)] \tilde{s} \) \( \pi(i) \notin S \) are mutually distinct and only appear on the RHS.

**Corollary 2.** Let \( \lambda \in \mathbb{N} \) be the security parameter and \( A \) be an adversary that on input \((1^\lambda, p)\) makes \( Q_{op} \) group operation queries to oracles \( O_{add} \) and \( O_{pair} \), as well as \( Q_{ct}, Q_{sk} \) queries to oracles \( O_{ct}, O_{sk} \), and \( Q_H \) queries to the random oracle \( H \). KP-ABE is adaptively secure in the GGM such that

\[ \text{Adv}_{\text{KP-ABE}, A}^\text{GGM}(\lambda) \leq \frac{3 \cdot (|S| + 2) \cdot Q_{ct} + (n_1 + 1) \cdot Q_{sk} + Q_{op})^2}{p} , \]

where \( |S| \) is the maximum size of the attribute sets queried to \( O_{ct} \) and \( n_1 \) is the maximum number of rows of \( M \) queried to \( O_{sk} \).

7 Implementation and Evaluation

We use two metrics to compare our scheme with prior work, the first is in terms of efficiency and the second is in terms of tightness.
7.1 Efficiency

We implemented several ABE schemes in Python 2.7.12 using the Charm 0.43 framework \[4\] and the MNT224 curve for pairings. We ran the schemes on a Lenovo Thinkpad Yoga X1 laptop with a 1.80GHz Intel Core i7-10510U CPU and 16GB RAM. Our implementation extends the code of Agrawal and Chase \[1\] and we provide the implementation on GitHub \[45\]. In particular, we compare the CP-ABE and KP-ABE schemes described in Table 1.

All schemes are implemented in the asymmetric setting. Agrawal and Chase already transferred the original constructions of BSW, Waters and GPSW that use symmetric bilinear maps to the asymmetric setting \[2, Appendices D-F\]. Apart from our schemes, we additionally implement the unbounded CP-ABE and KP-ABE of ABGW, see Appendix C for a self-contained description.

In our experiment, we use access policies of the form "Attr_1 and Attr_2 and ... and Attr_N" for \(N \in \{10, 20, \ldots, 100\}\) without re-use (i.e., \(\tau = 1\)). This way, \(|S| = n_1 = n_2 := N\) and all attributes are used in decryption. As \[2\], we first convert the policies into a Boolean formula and then to an MSP using the Lewko-Waters’ method \[39\]. This way, the matrix \(M\) has only entries in \(\{0, 1, -1\}\) and the reconstruction coefficients are always 0 or 1, reducing the number of exponentiations.

![Fig.3](image)

**Fig.3:** Running times for CP-ABE (top) and KP-ABE (bottom) schemes. We use one-use formulas (i.e., \(\tau = 1\)). In particular, for 100 attributes, CP-ABE decryption takes 0.016s in FABEO and 0.032s in FAME, and KP-ABE decryption takes 0.011s in FABEO and 0.031s in FAME.

In Figure 3, we show the average running times for the key generation, encryption and decryption algorithms. For additional reference, the average execution time for different group operations on the MNT224 curve is summarized in Table 2. All our experiments compute the average time in 20 executions. It is worth noting that each algorithm of our two schemes performs better or comparatively the same as all the others. These results are supported by our theoretical overview in Table 3 which lists the number of multiplications and exponentiations for each group as well as the number of hashing and pairing operations. Recall also that exponentiation in \(G_2\) is much slower than in \(G_1\) and the pairing operation is comparatively expensive. Additionally, we provide the number of group elements of secret keys and 10 The implementations in FAME and ABGW also use the Charm framework. Unfortunately, the PBC library used in Charm does not support BLS12-381.

18
ciphertexts in Table 5. Since in general elements in $G_2$ are about 2 to 3 times the size of elements in $G_1$, our keys and ciphertexts always achieve the same size or even improve considerably upon the other schemes. We provide a more detailed explanation on running times and sizes below.

**One-use restriction.** FAME has a one-use restriction described in [2, Section 4]. A common way to work around this problem is to make $\hat{\tau}$ copies of each attribute, for some $\hat{\tau}$ chosen at set-up; this way, FAME can support $\hat{\tau}$-use MSPs. The downside of this transformation is that in the CP-ABE, the size of the keys grow by a factor of $\hat{\tau}$ though encryption and decryption time are not affected. Similarly, in the KP-ABE, the ciphertexts and encryption time grow by a factor of $\hat{\tau}$. We explicitly account for $\hat{\tau}$ when describing FAME in our comparison tables. For applications where $\tau$ may be large and fast decryption is paramount, we can apply the same transformation to our schemes so that decryption only requires 2-3 pairings. For this reason, the experiments and most comparison in the text consider $\hat{\tau} = \tau = 1$.

A follow-up to FAME by Tomida, Kawahara and Nishimaki (TKN) [50] shows how to remove the one-use restriction using techniques from [37], paying a multiplicative factor $\tau$ in the number of pairings required for decryption, and a much larger security loss in the reduction to DLIN. The TKN scheme essentially coincides with FAME when $\tau = 1$, and for larger $\tau$, remains at least 2-3 times less efficient than FABEO. All of our experiments are for $\tau = 1$, hence the omission of TKN.

**Setup.** In Table 4, we show the setup time of all schemes listed in our evaluation. For schemes, where the universe size is bounded, we used the minimal bound $N$. Both our schemes have the shortest setup times (around 0.14s), however all schemes that support large universes are almost equally fast. The universe size of Waters and GPSW are bounded, thus the time increases with the universe size and for $N = 100$, they are about 3-7 times slower than the other schemes.

**CP-ABE.** Looking at running times of CP-ABE schemes given in Figure 3, only the key generation time of Waters CP-ABE can compete with ours, being only slightly faster (<0.01s for all test samples). This is due to the fact that the number of operations performed by Waters and our scheme are essentially the same. Whereas for our scheme it takes only 0.09s to generate a key for a set of 100 attributes, it takes at least 7 times longer for BSW, ABGW and FAME. This can be explained by that the fact that ABGW and BSW both perform exponentiations in $G_2$ that are linear in the number of attributes (cf. Table 3). Also it is easy to see that FAME is around 9 times slower than our scheme because of the number of exponentiations in $G_1$.

The encryption times vary a bit more, with our scheme being the fastest (0.18s for policies of 100 attributes), followed by ABGW and FAME, which are around 2 and 3 times slower. This also shows up in Table 3 and the number of exponentiations in $G_1$. BSW and Waters both perform $n_1$ number of exponentiations in $G_2$, therefore they are the least efficient schemes here.

FAME is the only other scheme that supports fast decryption which does not depend on the number of attributes. Without multi-use of attributes, our scheme is still 2 times faster than FAME. We can see this also in Table 3, since FAME performs 6 pairing operations and our scheme 3 (for $\tau = 1$). For both FAME and our scheme, decryption that uses 100 attributes takes less than 0.04s, whereas it takes almost 0.5s for ABGW and Waters. Those schemes need to perform a pairing operation for each attribute that is used in decryption.

**KP-ABE.** Looking at KP-ABE, the key generation time of our scheme is much lower than for all the others. Compared to GPSW and ABGW this can be explained by the fact that these two schemes perform

<table>
<thead>
<tr>
<th>Groups</th>
<th>Multiplication</th>
<th>Exponentiation</th>
<th>Hash</th>
<th>Pairing</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>.002</td>
<td>.638</td>
<td>.040</td>
<td>3.836</td>
</tr>
<tr>
<td>$G_2$</td>
<td>.017</td>
<td>4.717</td>
<td>12.342</td>
<td>-</td>
</tr>
<tr>
<td>$G_T$</td>
<td>.005</td>
<td>1.078</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 2:** Average time (in milliseconds) for operations on the MNT224 curve.
are also three times larger since they consist of more group elements. The ciphertext sizes of BSW and all group elements are in elements in secret keys of our CP-ABE scheme consist of ciphertexts. Note that in general, elements in Key and Ciphertext Sizes. In Table 5, we also show the number of group elements of secret keys and need to perform pairing operations that are linear in the number of attributes.

In our experiment) in experiments and most comparison in the text consider (counted with multiplicity). Note that \( \tau \leq I \). The experiments and most comparison in the text consider \( \tilde{\tau} = \tau = 1 \).

all exponentiations in \( G_2 \) instead of \( G_1 \), whereas our scheme only performs \( \tau \) (and therefore 1 operation in our experiment) in \( G_2 \). On the other hand, the number of exponentiations in FAME depends on \( n_1 \) and \( n_2 \) and thus is already around 6 times higher than for our scheme. Therefore, in our scheme it takes less than 0.2s for a policy size of 100, whereas GPSW needs more than half a second and FAME and ABGW more than a second. In terms of encryption time, GPSW is slightly below that of our scheme (around 0.01s), but both are around twice as fast as ABGW and considerably faster than FAME. This can be justified by the constant factors in the number of exponentiations in \( G_1 \) in Table 3. For decryption we get the same results as in the CP-ABE case since GPSW and ABGW do not support fast decryption and need to perform pairing operations that are linear in the number of attributes.

### Key and Ciphertext Sizes

In Table 5, we also show the number of group elements of secret keys and ciphertexts. Note that in general, elements in \( G_2 \) are about 2 to 3 times the size of elements in \( G_1 \). The secret keys of our CP-ABE scheme consist of \( m + 1 \) elements in \( G_2 \) and only 1 element in \( G_2 \), which is the same as in Waters. BSW uses fresh randomness for all attributes, thus their keys additionally include \( m \) elements in \( G_2 \). The number of group elements of keys in ABGW is the same as for our scheme, however all group elements are in \( G_2 \), which makes secret keys about three times larger. Secret keys in FAME are also three times larger since they consist of more group elements. The ciphertext sizes of BSW and Waters are exactly the same, requiring about the same number of elements in \( G_1 \) and \( G_2 \). Our scheme allows randomness reuse and therefore only needs \( \tau + 1 \) additional elements in \( G_2 \). ABGW and FAME both need around 3 times more elements than our scheme. Looking at KP-ABE, the advantage of our scheme is that most elements of the secret keys are in \( G_1 \), as is the case for FAME, which again requires around 3 times more elements. Secret keys in GPSW and ABGW contain only elements in \( G_2 \). The ciphertext of our scheme is comparable to that of GPSW and about 2 resp. 3 times smaller than those of ABGW and FAME.

### Table 3: Number of group operations in \( G_1 \) and \( G_2 \) for key generation and encryption of CP-ABE (top) and KP-ABE (bottom) schemes.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Key generation</th>
<th>Encryption</th>
<th>Decryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n_1 )</td>
<td>( m_1 + 1 )</td>
<td>( m + 2 )</td>
</tr>
<tr>
<td>BSW</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Waters</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FAME</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ABGW</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GPSW</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FAME</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ABGW</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 4: Setup times for CP-ABE (left) and KP-ABE (right) schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Uni size</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSW</td>
<td>-</td>
<td>0.025</td>
</tr>
<tr>
<td>Waters</td>
<td>100</td>
<td>0.096</td>
</tr>
<tr>
<td>ABGW</td>
<td>-</td>
<td>0.016</td>
</tr>
<tr>
<td>FAME</td>
<td>-</td>
<td>0.030</td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>0.014</td>
</tr>
<tr>
<td>GPSW</td>
<td>100</td>
<td>0.095</td>
</tr>
<tr>
<td>ABGW</td>
<td>-</td>
<td>0.015</td>
</tr>
<tr>
<td>FAME</td>
<td>-</td>
<td>0.030</td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>0.013</td>
</tr>
</tbody>
</table>
Similar to our work, ABGW uses a pair encoding approach and proves security in the hybrid argument on the number of ciphertexts as mentioned above. In particular, the security bounds in Table 6 are based on the security bounds stated in the corresponding papers as well as an additional evaluation. We omit Waters and GSPW here as those schemes are only selectively secure. The numbers may easily observe a large number of ciphertexts, rather than a large number of keys.

\[ Q_{ct} \leq 2^{256} \]

Table 5: Key and ciphertext sizes of CP-ABE (top) and KP-ABE (bottom) schemes. The columns \( G_1 \) and \( G_2 \) denote the number of elements in the respective group (in general, \( |G_2| \geq 2|G_1| \)). \( m \) denotes the number of attributes in the set \( S \), \( n_1 \) and \( n_2 \) are the number of rows and columns of the MSP matrix and \( \tau \) denotes the maximum number of multi-use. The experiments consider \( \tau = 1 \).

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Key size</th>
<th>Ciphertext size</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSW</td>
<td>( G_1 )</td>
<td>( G_2 )</td>
</tr>
<tr>
<td>Waters</td>
<td>( m + 1 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>FAME</td>
<td>( 3m + 3 )</td>
<td>( 3 )</td>
</tr>
<tr>
<td>ABGW</td>
<td>( - )</td>
<td>( m + 2 )</td>
</tr>
<tr>
<td>Ours</td>
<td>( m + 1 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>GPSW</td>
<td>( - )</td>
<td>( n_1 )</td>
</tr>
<tr>
<td>FAME</td>
<td>( 3n_1 )</td>
<td>( 3 )</td>
</tr>
<tr>
<td>ABGW</td>
<td>( - )</td>
<td>( 2n_1 )</td>
</tr>
<tr>
<td>Ours</td>
<td>( n_1 )</td>
<td>( \tau )</td>
</tr>
</tbody>
</table>

7.2 Bit Security based on Tightness

Whereas considering multiple secret key queries in the security definition is considered standard in terms of ABE security, we additionally consider many ciphertext or challenge queries in our security proof. The two definitions are polynomially equivalent, but the non-trivial implication from one to many ciphertexts incurs a security loss linear in the number of ciphertext queries. On the contrary, if the security loss is only constant, we say that the bound is tight, as is the case for our bounds. The security loss plays an important role in choosing the system parameters of the scheme, e.g., the size of the underlying pairing group which provides a determined level of security, which is usually stated in bits. Further, we can define the success ratio of an adversary \( A \) by its advantage \( \text{Adv} \) and its running time \( t \). For \( \lambda \)-bit security, we then require that \( \text{Adv}/t \leq 2^{-\lambda} \). From this value, we can then deduce whether a concrete instantiation provides the desired security level.

In Table 6, we compute the bit security of our scheme, as well as ABGW, FAME and BSW in different scenarios from small-scale to large-scale adversaries, based on the running time \( t \) captures the offline time of an adversary, e.g., to perform group operations or also to evaluate a hash function (thus including random oracle queries). We assume \( t \) to be rather large, whereas secret key and ciphertext queries are considered online running time and therefore considerably lower. The advantage also depends on the order of the underlying group and for our comparison we assume \( p = 2^{256} \). Since a discrete logarithm attack on the elliptic curve group yields a bound \( O(t^2/p) \), this parameter choice is based on a security level of around 128 bit and this should be the target for the bit security of the ABE schemes as well.

We consider four different scenarios from small-scale to large-scale adversaries, based on the running time \( t \in \{2^{40}, 2^{60}, 2^{80}, 2^{128}\} \). For each scenario, we choose the number of secret key queries \( Q_k \) and ciphertexts \( Q_{ct} \) accordingly, once for \( Q_{ct} = Q_k \) and once for \( Q_{ct} < Q_k \), since in practice an adversary may easily observe a large number of ciphertexts, rather than a large number of keys.

**Evaluation.** We omit Waters and GSPW here as those schemes are only selectively secure. The numbers in Table 6 are based on the security bounds stated in the corresponding papers as well as an additional hybrid argument on the number of ciphertexts as mentioned above. In particular, the security bounds that we use in our comparison in Table 6 can be explained as follows.

- **ABGW:** Similar to our work, ABGW uses a pair encoding approach and proves security in the the GGM. However, the use of rational fractions does not allow for the optimal bound and they get an advantage of \( O(Q_k Q_{ct} t^2/p) \), where the additional factor \( Q_{ct} \) comes from the hybrid argument.

- **FAME:** The schemes achieve security under the DLIN assumption using the dual system encryption framework, which also incurs a security loss in the number of secret keys. Since they also only consider the single-ciphertext setting, they achieve the same bound as ABGW, namely \( O(Q_k Q_{ct} t^2/p) \).
The scheme in BSW is proven secure in the GGM and single-ciphertext setting, thus there is no security loss in the number of secret keys. We get $O(\frac{2^{2n}t^2}{p})$ to account for many ciphertexts as well.

- BSW: The scheme in BSW is proven secure in the GGM and single-ciphertext setting, thus there is no security loss in the number of secret keys. We get $O(\frac{2^{2n}t^2}{p})$ to account for many ciphertexts as well.

- Ours: Since we prove security directly in the many-ciphertext setting and in the GGM, we achieve the optimal bound $O(\frac{t^2}{p})$.

Although all schemes meet the target bound in a small to medium-scale scenario (cf. Table 6), one can already observe the gap between the theoretical security level of ABGW or FAME and that of our schemes, which comes from their lose bounds. For $t = 2^{30}$, $Q_{sk} = 2^{40}$, both ABGW and FAME only provide a security level of $2^{116}$ bits, compared to the target of $2^{128}$. BSW and our scheme both still meet the target, however BSW is slightly below the target when increasing the number of ciphertexts to $2^{30}$, since their bound depends on $t$ and $Q_{ct}$. When allowing for a running time of $2^{128}$, then ABGW and FAME only provide 48 bits of security and therefore not suitable for applications in a large-scale scenario. BSW still achieves 78 resp. 88 bits, which may be sufficient for some applications. Due to the tight bound, our scheme exactly meets the target of 128 bits.

### Exact bounds

The numbers in Table 6 are computed based on the most dominating terms in the bounds as described above. In fact, when looking at our security statement, our bounds also depend on the number of secret keys and ciphertext and even their corresponding sizes (cf. terms $|ct|Q_{ct}$ and $|sk|Q_{sk}$ in Theorem 1). However, since $|ct|Q_{ct} + |sk|Q_{sk} \ll t$, these do not have a huge impact on the exact security of the schemes. For $n_1 = n_2 = 100$, we lose less than two bits of security. The same applies to BSW. Due to the rational fractions used in ABGW, the ciphertext and key sizes are actually an additional multiplicative factor in the bound, i.e., the dominating term is $|ct| \cdot |sk| \cdot Q_{ct}Q_{sk} t^2$. For large policies this has a considerable impact on the security. In this case, we lose 19 bits of security when $n_1 = n_2 = 100$. Compared to the values in Table 6, a small security loss of around 3 bits also occurs in FAME, which is due to constant factors.

### 8 Extensions

In this section, we briefly describe how we can extend our definition of PES-ABE to capture more schemes, e.g., ABE for deterministic finite automata (DFA).
8.1 A variant of PES-ABE

We want to capture PES-ABE schemes as in ABGW with Setup, Enc0, KeyGen0 as before, except:

\[
\begin{align*}
\text{mpk} & := ([b]_1, [\alpha]_\mathcal{P}), \\
\text{msk} & := (b, \alpha), \\
\text{ct} & := ([c^1]_1, [c^2]_1), \\
\text{sk} & := ([k^1]_2, [k^2]_2)
\end{align*}
\]

For such schemes, we impose an additional constraint on \(k^1, k^2\) as with ABGW, namely that \(r = (r'||r'')\) and \(k^1[\alpha, r', b \otimes r''), k^2(r')\) (that is, we removed \(r', b \otimes r''\) from the input to \(k^1\) and \(r''\) from the input to \(k^2\))\footnote{ABGW refers to \(r'\) as the non-lone variables and \(r''\) as the lone variables. Also, ABGW considers a more general setting for \(c^1, c^2\) with \(s = (s'||s'')\) and \(c^2(s'), c^1(s'', b \otimes s')\). To the best of our knowledge, none of the existing ABE schemes exploit this generalization.}. This way, we can ensure that \(\text{span}(k^1) \cap \text{span}(k^2) = \{0\}\), which we will need in the proof of strong symbolic security.

**Strong Symbolic Security (Variant).** For all \(Q_{\text{ct}}, Q_{\text{sk}} \in \mathbb{N}, X \in \mathcal{X}^{Q_{\text{ct}}}, Y \in \mathcal{Y}^{Q_{\text{sk}}}\) such that \(P(X[i], Y[j]) = 0\) for all \(i \in [Q_{\text{ct}}], j \in [Q_{\text{sk}}]\), we have

\[
\text{span}(\bar{\alpha} \otimes c_X^1) \cap \text{span}(\bar{\alpha} \| (1||\bar{b})||c_X^1 \otimes (1||k_Y^1||k_Y^2)) = \{0\},
\]

where \(X, Y, c_X^1, c_X^2, k_Y^1, k_Y^2\) are as in Definition\footnote{ABGW refers to \(r'\) as the non-lone variables and \(r''\) as the lone variables. Also, ABGW considers a more general setting for \(c^1, c^2\) with \(s = (s'||s'')\) and \(c^2(s'), c^1(s'', b \otimes s')\). To the best of our knowledge, none of the existing ABE schemes exploit this generalization.}.

**Claim.** If PES-ABE satisfies \((1,1)\)-symbolic security (Definition\footnote{ABGW refers to \(r'\) as the non-lone variables and \(r''\) as the lone variables. Also, ABGW considers a more general setting for \(c^1, c^2\) with \(s = (s'||s'')\) and \(c^2(s'), c^1(s'', b \otimes s')\). To the best of our knowledge, none of the existing ABE schemes exploit this generalization.}), then it also satisfies the variant of strong symbolic security.

**Proof.** The proof is similar to that of Lemma\footnote{ABGW refers to \(r'\) as the non-lone variables and \(r''\) as the lone variables. Also, ABGW considers a more general setting for \(c^1, c^2\) with \(s = (s'||s'')\) and \(c^2(s'), c^1(s'', b \otimes s')\). To the best of our knowledge, none of the existing ABE schemes exploit this generalization.} Step 1 and step 2 already tell us that \((1,1)\) symbolic security implies that

\[
\text{span}(\bar{\alpha} \otimes c_X^1) \cap \text{span}(\bar{\alpha} \| (1||\bar{b})||c_X^1 \otimes (1||k_Y^1||k_Y^2)) = \{0\}.
\]

We need to slightly modify the last step of the proof. We also prove the claim by contradiction. So assume there exist \(e^*, e^{(1)}, e^{(2)}, e^{(3)}\) such that

\[
(\bar{\alpha} \otimes c_X^2) \cdot e^{(1)} = (\bar{\alpha} \| c_X^1)(1||\bar{b})\|e_X^1 \otimes (1||k_Y^1||k_Y^2)) \cdot e^{(1)} + (c_X^1 \otimes k_Y^1||c_X^2 \otimes k_Y^2) \cdot e^{(2)}
\]

In the same way as in the proof of Lemma\footnote{ABGW refers to \(r'\) as the non-lone variables and \(r''\) as the lone variables. Also, ABGW considers a more general setting for \(c^1, c^2\) with \(s = (s'||s'')\) and \(c^2(s'), c^1(s'', b \otimes s')\). To the best of our knowledge, none of the existing ABE schemes exploit this generalization.}, we can show that the first term evaluates to 0. It remains to show that also the second term evaluates to 0, which follows from the fact that

\[
\text{span}(\bar{\alpha} \otimes c_X^1) \cap \text{span}(c_X^1 \otimes k_Y^1||c_X^2 \otimes k_Y^2) = \{0\}
\]

together with

\[
\text{span}(c_X^1 \otimes k_Y^1||c_X^2 \otimes k_Y^2) \cap \text{span}(c_X^1 \otimes k_Y^1||c_X^2 \otimes k_Y^2) = \{0\}
\]

The first equation holds since \(\text{span}(c_X^1 \otimes k_Y^1)\) contains monomials of the form \((\bar{s}_X \otimes \bar{b}) \| (\alpha || r_Y^1 \| \bar{b} \otimes r_Y^2)\) and \(\text{span}(c_X^2 \otimes k_Y^2)\) contains monomials of the form \(\bar{s}_X \otimes r'\), which all do not appear in \(\text{span}(\bar{\alpha} \otimes c_X^2)\).

The second equation holds since \(\text{span}(c_X^1 \otimes k_Y^1)\) contains monomials of the form \(\bar{s}_X \otimes \bar{b} \otimes r_Y^2\) and \(\text{span}(c_X^2 \otimes k_Y^2)\) contains monomials of the form \(\bar{s}_X \otimes (\alpha || r_Y^1 \| \bar{b} \otimes r_Y^2)\), which all do not appear on the RHS of \(\cap\). Note that here we use that \(\text{span}(r') \cap \text{span}(r'') = 0\).

8.2 ABE for DFA

We consider the ABE scheme for DFAs in \cite{1} (building on \cite{2}). Recall that a DFA is specified by a tuple \((Q, \Sigma, \delta, F)\) where the state space is \([Q] := \{1, 2, \ldots, Q\}; 1\) is the unique start state; \(F \subseteq [Q]\) is the set of accept states, and \(\delta : [Q] \times \Sigma \rightarrow [Q]\) is the state transition function.

We provide a self-contained overview of our ABE scheme for DFA in Figure\footnote{ABGW refers to \(r'\) as the non-lone variables and \(r''\) as the lone variables. Also, ABGW considers a more general setting for \(c^1, c^2\) with \(s = (s'||s'')\) and \(c^2(s'), c^1(s'', b \otimes s')\). To the best of our knowledge, none of the existing ABE schemes exploit this generalization.} In the following, we describe the underlying PES-ABE.
Setup($1^\lambda$). Run GroupGen($1^\lambda$) to obtain $G := (g, G_1, G_2, G_T, e, g_1, g_2)$. Pick $u, w_{\text{start}}, w_{\text{end}}, z \in \mathbb{Z}_p$, $\{w_x\}_{x \in \Sigma} \in \mathbb{Z}_p^{\Sigma}$. Compute the master public key as

$$\text{mpk} := (G, g, g_1^{w_{\text{start}}}, g_1^{w_{\text{end}}}, g_1, \{g_2^u\}_{u \in \Sigma}, e(g_1, g_2)^w)$$

Let $\text{msk} := (a, w_{\text{start}}, w_{\text{end}}, z, \{w_x\}_{x \in \Sigma})$ be the master secret key.

KeyGen($\text{msk}, (Q, \Sigma, \delta, F)$). Pick $\{r_u\}_{u \in \{0\}} \in \mathbb{Z}_p^Q$, $\{d_u\}_{u \in \{0\}} \in \mathbb{Z}_p^Q$. Compute

$$sk_0 := g_2^{a+w_{\text{start}}}, \text{sk}_{1,u} := g_2^{x^u},$$

for all $u \in \{0\}$. Then compute

$$\text{sk}_{2,u} := g_2^{-d_u+zr_u}, \text{sk}_{3,u,\sigma} := g_2^{-d_u+wr_u},$$

for all $u \in \{0\}, \sigma \in \Sigma$, where $v = \delta(u, \sigma)$, and

$$\text{sk}_{4,u} = g_2^{-d_u+w_{\text{end}}r_u}$$

for all $u \in F$. Output $\text{sk} := (sk_0, (sk_{1,u})_{u \in \{0\}}, (sk_{2,u}, sk_{3,u,\sigma})_{u \in \{0\}, \sigma \in \Sigma}, (sk_{4,u})_{u \in F})$.

Enc($\text{mpk}, x$). Pick $(s_0, s_0, s_1, \ldots, s_{\ell-1}) \in \mathbb{Z}_p^{\ell+1}$, where $\ell = |x|$. Compute

$$\text{ct}_0 := g_1^{s_0}, \text{ct}_1 := g_1^{s_0 w_{\text{start}}}, \text{ct}_{2,i} := g_1^{s_i}, \text{ct}_{3,i} := g_1^{s_{i-1}+s_i z r_i}, \text{ct}_4 := g_1^{s_{\ell-1} w_{\text{end}}},$$

for all $i \in \llbracket \ell \rrbracket$. Output $\text{ct} := (\text{ct}_0, \text{ct}_1, (\text{ct}_{2,i}, \text{ct}_{3,i})_{i \in \llbracket \ell \rrbracket}, \text{ct}_4)$ and $d := e(g_1, g_2)^{\alpha s}.

Dec($\text{mpk}, x, (Q, \Sigma, \delta, F), \text{ct}, \text{sk}$). If $(Q, \Sigma, \delta, F)$ accepts $x$, then there exists a sequence $u_1, \ldots, u_{\ell} \in Q$ such that $u_1 = 1$, $\delta(u_1, x_1) = u_{i+1}$ for all $i \in \llbracket \ell-1 \rrbracket$ and $\delta(u_{\ell}, x_{\ell}) \in F$. Compute

$$B := e(\text{ct}_0, \text{sk}_0) \cdot \prod_{i \in \llbracket \ell \rrbracket} e(\text{ct}_{2,i}, \text{sk}_{2,i,u}) \cdot e(\text{ct}_{3,i}, \text{sk}_{3,i,u,\sigma}) \cdot e(\text{ct}_4, \text{sk}_{4,u,\sigma})$$

and output

$$B' := e(\text{ct}_{2, \ell}, \text{sk}_{4, \ell}(u_{\ell}, \sigma)) / e(\text{ct}_{4, \ell}(u_{\ell}, \sigma)).$$

Fig. 4: Our KP-ABE scheme for DFA.

- Setup$_0$. Output $n := 3 + |\Sigma|$, where we parse $b$ as $(w_{\text{start}}, w_{\text{end}}, z, \{w_x\}_{x \in \Sigma})$.

- Enc$_0(x)$. Set $w = \ell + 1$, $w_1 = \ell + 2$, $w_2 = \ell + 1$, and output $(c_1, c_2)$ where we parse $s$ as $(s_0, s_0, s_1, \ldots, s_{\ell-1})$ and

$$c_1(s \otimes b) := (s_0 w_{\text{start}} \parallel \{s_{i-1} z + s_i z r_i\}_{i \in \llbracket \ell \rrbracket} \parallel s_f w_{\text{end}}),$$

$$c_2(s) := (s).$$

- KeyGen$_0(Q, \Sigma, \delta, F)$. Set $m = 2Q$, $m_1 = 1 + Q + Q \cdot |\Sigma| + |F|$, $m_2 = Q$, and output $(k_1, k_2)$ where we parse $r = (r'|r'') := (\{r_u\}_{u \in \{0\}} \parallel \{d_u\}_{u \in \{0\}})$ and

$$k_1(\alpha, r'' \otimes r') := (d_1 + w_{\text{start}} r_1 \parallel \{d_u + z r_u\}_{u \in \{0\}} \parallel \{d_{\delta(u, \sigma)} + w_{\sigma} r_u\}_{u \in \{0\}, \sigma \in \Sigma} \parallel \{\alpha - d_u + w_{\text{end}} r_u\}_{u \in F})$$

$$k_2(r') := (r').$$

In applications, think of $\ell \gg |\Sigma|, Q$. We note that our scheme differs from Waters’ scheme in that we reuse $r_u$ for all the transitions starting from $u$ instead of a fresh $r_{u, \sigma}$ for each $(u, \sigma)$. This modification yields a smaller secret key (cf. Table 7).

We now prove correctness and symbol security.

Correctness. Let $ct$ be a ciphertext for $x$ and $sk$ be a secret key for $(Q, \Sigma, \delta, F)$ as defined in Figure 4. If $(Q, \Sigma, \delta, F)$ accepts $x$, then there exists a sequence $u_1, \ldots, u_{\ell} \in Q$ such that $u_1 = 1$, $\delta(u_1, x_1) = u_{i+1}$ for all $i \in \llbracket \ell-1 \rrbracket$ and $\delta(u_{\ell}, x_{\ell}) \in F$. Then decryption computes

$$(1) \frac{e(\text{ct}_0, \text{sk}_0)}{e(\text{ct}_1, \text{sk}_{1,1})} = [s_0(d_1 + w_{\text{start}} r_1)]_T = [s_0 d_1]_T$$
Claim 1 basically tells us that we can ignore all terms in \( t \). In order to prove claim 2, we want to look at the monomials on the RHS. The definition of the scheme shows that:\n
\begin{align*}
\frac{e(\alpha_1,\delta,\omega) \cdot e(\alpha_2,\delta,\omega)}{e(\alpha_1,\delta,\omega)} &= [s_{i-1}(-d_{u_i} + zw_{u_i}) + s_i(d_{s(u_i,\sigma)} + w_{x_i}r_{u_i}) - (s_{i-1}z + s_1w_{x_i})r_{u_i}]_T \\
&= [s_id_{s(u_i,\sigma)} - s_{i-1}d_{u_i}]_T
\end{align*}

(2)

To prove claim 1, we look at the monomials on the RHS of Step 2.\n
\begin{align*}
\frac{e(\alpha_1,\delta,\omega) \cdot e(\alpha_2,\delta,\omega)}{e(\alpha_1,\delta,\omega)} &= [s_{i-1}(-d_{u_i} + zw_{u_i}) + s_i(d_{s(u_i,\sigma)} + w_{x_i}r_{u_i}) - (s_{i-1}z + s_1w_{x_i})r_{u_i}]_T \\
&= [s_id_{s(u_i,\sigma)} - s_{i-1}d_{u_i}]_T
\end{align*}

(3)

Correctness follows from the fact that \( \alpha \cdot s_i \cdot s \cdot 0 \equiv 0 \).\n
We now proceed in three steps and make the following claims.\n
**W.l.o.g.** one can assume that \( |F| = 1 \), i.e., there is only one accepting state and that it has no outgoing transitions. In the following, we will denote the accepting state by \( \alpha \). Further, we will denote the sequence of states on input \( x \) by \( (u_1, \ldots, u_\ell, \text{fin}) \), where \( u_1 = 1 \) is the start state, \( u_{i+1} = \delta(u_i, x_i) \) for \( i \in [\ell] \) is the state reached upon reading \( x_1, \ldots, x_i \), and \( \text{fin} = \delta(u_\ell, x_\ell) \) is the final state.\n
We will prove the above claim by contradiction. Thus, assume on the contrary that:\n
\[
\text{span}(\alpha \otimes c^2) \cap \text{span}(c^1 \otimes k^2 \| k^1 \otimes c^2) = \{0\}.
\]

We now proceed in three steps and make the following claims.\n
**Claim 1**: \( \text{span}(\alpha \otimes (s_\ell_0) \cdots \| s_{i-1}) \cap \text{span}(c^1 \otimes k^2 \| k^1 \otimes c^2) = \{0\} \).\n
**Claim 2**: If \( \alpha s_\ell \in \text{span}(c^1 \otimes k^2 \| k^1 \otimes c^2) \), then:\n
\[
\bar{s}_i d_{\alpha c} \in \text{span}(s_\ell d_1 \{s_i d_{s(u_i,\sigma)} - s_{i-1}d_{u_i}\}_{i \in [\ell], u \in [Q]})
\]

**Claim 3**: \( \bar{s}_i d_{\alpha c} \notin \text{span}(s_\ell d_1 \{d_{s_i d_{s(u_i,\sigma)} - s_{i-1}d_{u_i}}\}_{i \in [\ell], u \in [Q]}) \).

Claim 1 basically tells us that we can ignore all terms in \( \text{span}(\alpha \otimes c^2) \) except for \( \alpha s_\ell \). Claim 2 shows that the only degree two monomials we learn from \( \text{span}(c^1 \otimes k^2 \| k^1 \otimes c^2) \) are linear combinations of the terms \( s_\ell d_{s(u_i,\sigma)} - s_{i-1}d_{u_i} \). Claim 3 then yields a contradiction to claims 1 and 2, thus proving symbolic security.

**Step 1**: To prove claim 1, we look at the monomials on the RHS of \( \cap \). Note that the terms \( \alpha s_i \) for \( i \in \{0, \ldots, \ell-1\} \) only appear together with \( \bar{w}_{end} \bar{r}_x \bar{s}_i \). However, these monomials do not appear anywhere else, since \( \bar{w}_{end} \) only appears in \( \bar{w}_{end} \bar{r}_x \bar{s}_i \) and the above equation must hold.

**Step 2**: Assume\n
\[
\alpha s_\ell \in \text{span}(c^1 \otimes k^2 \| k^1 \otimes c^2)
\]

In order to prove claim 2, we want to look at the monomials on the RHS. The definition of the scheme tells us that:\n
\[
\text{span}(c^1 \otimes k^2) = \text{span}\{(\bar{w}_{start} \bar{s}_0 \bar{r}_u \bar{w}_{end} \bar{s}_i \bar{r}_u)_{u \in [Q]}\}\{(\bar{s}_{i-1} + \bar{w}_x \bar{s}_i)\bar{r}_u\}_{u \in [Q], i \in [\ell]}\}
\]

and\n
\[
\text{span}(k^1 \otimes c^2) = \text{span}\{(\bar{w}_x + \bar{r}_{start} \bar{s}_\ell) \otimes (\bar{s}_0 \cdots \| \bar{s}_{i-1} \| \bar{s}_i)\}(\bar{d}_1 + \bar{w}_{start} \bar{s}_i \bar{r}_u) \otimes (\bar{s}_0 \bar{s}_1 \cdots \| \bar{s}_i) \}
\]

\[
\{(\bar{d}_u + \bar{r}_u \bar{d}_{s(u,\sigma)} \| \bar{w}_{\sigma} \bar{r}_u)_{u \in [Q], \sigma \in [\Sigma]} \}
\]

**Table 7**: Key and ciphertext sizes of ABE schemes for DFA.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Key size</th>
<th>Ciphertext size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waters</td>
<td>- 3</td>
<td>(</td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>(</td>
</tr>
</tbody>
</table>

\[
(2)
\]

\[
(3)
\]

\[
(8)
\]
where terms that are underlined will be removed in the next step. We also show that some of the terms that are underlined with a wavy line can also be removed. We will explain the details in the next paragraph.

In particular, we first look at the terms containing \( \tilde{w}_{\text{start}} \) and \( \tilde{w}_{\text{end}} \). Only the monomial \( \tilde{w}_{\text{start}} \tilde{r}_1 \tilde{s}_0 \) appears more than once, so we can remove all terms with \( \tilde{w}_{\text{start}} \tilde{r}_1 \tilde{s}_0 \) for \( i \in [\ell], u \in [Q] \setminus \{1\} \) since their coefficients must be 0. Also note that we can ignore \( \tilde{w}_{\text{start}} \tilde{r}_1 \tilde{s}_0 \) in \( (\tilde{d}_1 + \tilde{w}_{\text{start}} \tilde{r}_1) \tilde{s}_0 \) and in span \((c^I \otimes k^I)\) since their coefficients must always match to cancel out. The same applies for \( \tilde{w}_{\text{end}} \tilde{r}_1 \tilde{s}_0 \). Monomials \( \tilde{w}_{\text{end}} \tilde{r}_u \tilde{s}_i \) for \( u \in [Q] \setminus \{ac\} \) and \( i \in [0, \ell - 1] \) do not appear anywhere else and we can also ignore \( \tilde{w}_{\text{end}} \tilde{r}_u \tilde{s}_0 \) since the monomials containing \( \tilde{w}_x \) and \( \tilde{z} \) only appear on the RHS of (9) and only together with \( \tilde{s}_t \tilde{d}_u \). From (9) and the observation above, we get

\[
\tilde{s}_t \tilde{d}_u \in \text{span}(\tilde{s}_0 \tilde{d}_1 \| \{\tilde{s}_t \tilde{d}_{u, x, i} - \tilde{s}_i - 1 \tilde{d}_u\} \in [\ell], u \in [Q]) \tag{10}
\]

which proves claim 2.

Step 3. We prove the final claim by contradiction. Assume on the contrary that

\[
\tilde{s}_t \tilde{d}_u \in \text{span}(\tilde{s}_0 \tilde{d}_1 \| \{\tilde{s}_t \tilde{d}_{u, x, i} - \tilde{s}_i - 1 \tilde{d}_u\} \in [\ell], u \in [Q]) \tag{10}
\]

Then we can rewrite (10) as

\[
\tilde{s}_t \tilde{d}_u \in \text{span}(\tilde{s}_0 \tilde{d}_1 \| \{\tilde{s}_t \tilde{d}_{u, x, i} - \tilde{s}_i - 1 \tilde{d}_u\} \in [\ell], u \in [Q]) 
\]

We now argue that all coefficients of elements on the RHS must be 0 and thus we get a contradiction to (10). We look at the monomials that appear in the above equation. Since \( \tilde{s}_0 \tilde{d}_u \) for \( u \in [Q] \) only appears on the RHS, the coefficients of \( \{\tilde{s}_t \tilde{d}_{u, x, i} - \tilde{s}_i - 1 \tilde{d}_u\} \in [Q] \) must be 0. However, now the only term where \( \tilde{s}_1 \tilde{d}_u \) appears is \( (\tilde{s}_2 \tilde{d}_{u, x, i}) - \tilde{s}_1 \tilde{d}_u \), so these coefficients must also be 0 for all \( u \in [Q] \). We can go on with this argument until we reach the terms \( \tilde{s}_t \tilde{d}_{u, x, i} - \tilde{s}_i - 1 \tilde{d}_u \). Obviously, the coefficients of these terms must also be 0 and thus we get

\[
\tilde{s}_t \tilde{d}_u \in \text{span}(\tilde{s}_0 \tilde{d}_1) 
\]

However, this yields a contradiction to the fact that \( P(x, (Q, \Sigma, \delta, F)) = 0 \) since \( ac \neq 0 \), which concludes the proof of symbolic security.

Acknowledgements

Doreen Riepel was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2092 CASA - 390781972 and the European Union (ERC AdG REWORC - 101054911). Part of this work was done during an internship at NTT Research and a visit at UC Berkeley. We would also like to thank Kei Karasawa and his team for motivating discussions, Sanjam Garg for hosting Doreen Riepel at UC Berkeley, as well as Romain Gay for sharing his manuscript [24].
References


A Attribute-based Encryption in the PKE Setting

Many-Ciphertext CPA Security. We define security by a game between a challenger and an adversary $B$. The challenger picks a random challenge bit $b$ and provides the following oracles to $B$.

- Setup oracle $O_{mpk}$: This oracle can only be queried once and it must be the first query. The challenger runs Setup to obtain $(msk, mpk)$ and outputs $mpk$ to $B$.

- Ciphertext (or challenge) oracle $O_{ct}$: On the $i$-th query, $B$ provides $x_i \in \mathcal{X}$ as well as two messages $(m_i^{(0)}, m_i^{(1)})$ of equal length. The challenger runs $ct_i \leftarrow Enc(mpk, x_i, m_i^{(j)})$ and outputs $ct_i$.

- Secret key oracle $O_{sk}$: On the $j$-th query, $B$ provides $y_j \in \mathcal{Y}$. The challenger runs $sk_j \leftarrow KeyGen(msk, y_i)$ and outputs $sk_j$.

Finally, $B$ outputs a bit $b'$. We say that $B$ wins the game if $b = b'$ and for all queries $x_i$ and $y_j$, either $P(x_i, y_j) = 0$ or $m_i^{(0)} = m_i^{(1)}$. Note that this ensures that $B$ cannot win trivially.

**Definition 4.** An ABE scheme is adaptively many-ciphertext secure in the PKE setting if for all efficient $B$

$$\text{Adv}_{\text{ABE},B}(\lambda) := \left| \Pr[b = b'] - \frac{1}{2} \right|$$

is negligible in $\lambda$.

It is commonly known that the security notions in the KEM and PKE setting are equivalent. For completeness, we briefly sketch the proof that many-challenge KEM security as used in the main body of this paper tightly implies the many-ciphertext CPA security as defined above. Let $B$ be an adversary against many-ciphertext CPA security in the PKE setting. We construct a reduction $A$ against many-challenge KEM security. $A$ draws a random challenge bit $b \xleftarrow{} \{0,1\}$ and then runs adversary $B$. It answers $B$’s queries as follows.

- When $B$ queries $O_{mpk}$, $A$ queries its own $O_{mpk}$ oracle to receive the master public key $mpk$ which it forwards to $B$.

- When $B$ issues a ciphertext query $(x_i, m_i^{(0)}, m_i^{(1)})$, where $m_i^{(0)} \neq m_i^{(1)}$, $A$ queries its own challenge oracle on $x_i$ to receive an encapsulation $c_i$ and challenge key $d_{i}^{(\beta)}$, where $\beta$ is the challenge bit of the KEM security experiment. It outputs $ct_i = (c_i, d_{i}^{(\beta)} \oplus m_i^{(b)})$ as ciphertext.

- When $B$ issues a ciphertext query of the form $(x_i, m_i, m_i)$, $A$ runs the encapsulation algorithm on its own using the master public key to compute a pair $(c_i, d_i)$ and outputs $ct_i = (c_i, d_i \oplus m_i)$.

- When $B$ issues a secret key query $y_j$, $A$ forwards this query to its own secret key oracle and returns the result back to $B$. 29
Note that queries of the third type are the only possibility for \( \mathcal{B} \) to issue queries such that \( P(x_i, y_j) = 1 \) for some secret key query \( y_j \). Since \( \mathcal{A} \) does not query any of its own oracles at this point, it behaves correctly.

Finally \( \mathcal{B} \) outputs a bit \( b' \) and if \( b' = b \), \( \mathcal{A} \) outputs \( b' = 0 \) (real key). Otherwise, it outputs \( b' = 1 \) (random key).

We analyze the success probability by distinguishing two cases:

- Case \( \beta = 0 \): In this case \( \mathcal{A} \) always receives the real KEM key and it perfectly simulates the many-ciphertext CPA game for \( \mathcal{B} \). Thus, \( \Pr[\beta = \beta' | \beta = 0] = \text{Adv}_{\text{ABE}, \mathcal{B}}(\lambda) + 1/2 \).

- Case \( \beta = 1 \): In this case \( \mathcal{A} \) always receives a random KEM key. Thus, the challenge ciphertexts do not leak any information about \( \beta \). Also queries of the second type are independent of \( \beta \). We get \( \Pr[\beta = \beta' | \beta = 1] = 1/2 \).

Collecting the probabilities yields \( \text{Adv}_{\text{ABE}, \mathcal{B}}(\lambda) \leq 2 \cdot \text{Adv}_{\text{ABE}, \mathcal{A}}(\lambda) \).

## B Selective Security from \( q \)-type Assumptions

We can prove selective security of our CP-ABE and KP-ABE scheme (in the standard one-ct, many-sk setting) under a \( q \)-type assumption. In the selective security setting, the adversary must first commit on a challenge \( x^* \). We recall the formal definition below.

**Selective CPA Security.** We define security by a game between a challenger and an adversary \( \mathcal{A} \). At the beginning of the experiment, \( \mathcal{A} \) chooses a challenge \( x^* \) and sends it to the challenger. Then the challenger picks a random challenge bit \( \beta \) and provides the following oracles to \( \mathcal{A} \):

- Setup oracle \( O_{\text{mpk}} \): This oracle can only be queried once and it must be the first query. The challenger runs Setup to obtain (msk, mpk) and outputs mpk to \( \mathcal{A} \).

- Ciphertext (or challenge) oracle \( O_{\text{ct}} \): The challenger runs (ct, d(0)) \( \leftarrow \text{Enc}(\text{mpk}, x^*) \), chooses a random key \( d^{(1)} \in K \) and outputs (ct, d(\( \beta \))).

- Secret key oracle \( O_{\text{sk}} \): On the \( j \)-th query, \( \mathcal{A} \) provides \( y_j \in \mathcal{Y} \). The challenger runs sk \( \leftarrow \text{KeyGen}(\text{msk}, y_j) \) and outputs sk \( \mathcal{J} \).

We consider the single-challenge setting here, which means that \( O_{\text{ct}} \) can only be queried once, whereas \( O_{\text{sk}} \) can be queried adaptively and an arbitrary polynomial number of times. Finally, \( \mathcal{A} \) outputs a bit \( \beta' \).

We say that \( \mathcal{A} \) wins the game if \( \beta = \beta' \) and \( P(x^*, y_j) = 0 \) for all queries \( y_j \).

**Definition 5.** An ABE scheme is selectively secure if for all efficient \( \mathcal{A} \),

\[
\text{Adv}_{\text{ABE}, \mathcal{A}}^\text{selective}(\lambda) := \left| \Pr[\beta = \beta'] - \frac{1}{2} \right|
\]

is negligible in \( \lambda \).

**Our \( q \)-type assumption.** Our new assumption is parameterized by \( n \). It allows us to prove security of both schemes based on the same assumption but for different parameters \( n \) (see Appendices B.1 and B.2). In Appendix B.3 we show how to relate our assumption to another assumptions used in the literature which in turn has been analyzed in the GGM. Our new assumption is defined below.

**Definition 6 (n-q-type Assumption).** Let \( \mathcal{G} = (p, G_1, G_2, G_T, e, g_1, g_2) \) be a pairing group. Pick \( (x||y) \leftarrow \mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{Z}_p^n \). Compute

\[
D^1 = \left\{ g_1, g_1^x, \left\{ g_1^{y[i]} \right\}_{i \in [n]}, \left\{ g_1^{y[i]} \right\}_{i,j \in [n], i \neq j} \right\},
\]

\[
D^2 = \left\{ g_2, g_2^x, \left\{ g_2^{y[i]} \right\}_{i \in [n]} \right\}
\]

and \( T_0 = e(g_1, g_2), T_1 \leftarrow G_T \). The \( n \)-q-type assumption says that

\[
\text{Adv}_{\text{ABE}, n}(\lambda) := \left| \Pr[D(p, G_1, G_2, G_T, e, D^1, D^2, T_0) \Rightarrow 1] - \Pr[D(p, G_1, G_2, G_T, e, D^1, D^2, T_1) \Rightarrow 1] \right|
\]

is negligible in \( \lambda \).
B.1 CP-ABE

We prove security of our CP-ABE scheme (cf. Figure 1) for $\tau = 1$ in the random oracle model under the above q-type assumption for $n = n_2$, where $n_2$ is the number of columns of $M$ used in the challenge query.

Theorem 2. Let $\lambda$ be the security parameter. For any adversary $A$ against selective security of CP-ABE with $\tau = 1$ that issues at most $Q_{sk}$ queries to $O_{sk}$, one query to $O_{ct}$ and at most $Q_{H}$ queries to the random oracle $H$, there exists an adversary $D$ such that

$$
\text{Adv}_{\text{selective}}^{\text{CP-ABE},A}(1^\lambda) \leq \text{Adv}_{\text{q-type}}^{\text{CP-ABE},A}(1^\lambda) + \frac{1}{p},
$$

where $n_2$ is the number of columns of $M$ used in the challenge query.

Proof. We prove the theorem by reduction. Let $A$ be an adversary against selective security of the KP-ABE scheme in Figure 1. We construct an adversary $D$ against our q-type assumption that simulates the security game for $A$.

$D$ inputs $(p, G_1, G_2, G_T, e, D^1, D^2, T_b)$ and runs adversary $A$. First $A$ commits on a challenge $(M, \pi)$. Now $D$ must simulate the first query to $O_{mpk}$, the challenge query to $O_{ct}$ and adaptive queries to $O_{sk}$ and $H$.

Simulating $O_{mpk}$. We implicitly set $\alpha = \frac{\epsilon}{2}$. Output public parameters

$$
(p, G_1, G_2, G_T, e, g_1, g_2^{\frac{\epsilon}{2}}, e(g_1, g_2^{\frac{\epsilon}{2}})^{\alpha})
$$

where the last term can be computed by $e(g_1^{\frac{1}{\lambda}}, g_2^{[1]}).

Simulating $H$. We simulate the random oracle by lazy sampling and using a list $H$ to store all (distinct) queries and outputs. In particular, the list contains entries of the form $(u, H(u))$, where $u$ is the string (attribute) queried to $H$ and $H(u)$ is a group element. On each query, we first check if $u$ is contained in the list and if this is the case, answer consistent to the previous query. Otherwise, we will simulate the output as follows. First pick a new exponent $e_u \in \mathbb{Z}_p$ which will be used for randomization. Then we distinguish three cases and compute

$$
H(u) := \begin{cases} 
    g_1^{-\frac{1}{\lambda} |\mathcal{U}|} & \text{if } u = |\mathcal{U}| + 1 \\
    \sum_{i \in [n_2]} m_{\pi(i)} \cdot g_1^{e_u} & \text{if } \exists i \text{ s.t. } \pi(i) = u \\
    g_1^{e_u} & \text{otherwise}
\end{cases}
$$

Then add $(u, H(u))$ to the list and output $H(u)$. Note that we can compute all the terms using $g_1, g_1^{-\frac{1}{\lambda} |\mathcal{U}|}$ provided by the assumption.

Simulating $O_{ct}$. To compute the challenge for $(M, \pi)$ provided at the beginning, we (implicitly) set $s_1 = x + e_1$, $s' = x + e'$ and $v[j] = v(j) \cdot g_2^{[1]} + e_j$ for $j \in [2, n_2]$.

$$
\text{ct}_1 := g_2^{s} \cdot g_2^{s'}
$$

$$
\text{ct}_2 := g_2^{s} \cdot g_2^{s'}
$$

$$
\text{ct}_{3,i} := g_1^{m_{\pi(i)} \cdot v[j]} \cdot \left(\sum_{j \in [n_2]} m_{\pi(j)} \cdot g_1^{e_j}\right)^{s'}
$$

$$
d := T_b \cdot e(g_1^{\frac{1}{\lambda} |\mathcal{U}|}, g_2^{[1]} e_1)
$$
It is easy to see that we can simulate $c_{t_1}$ and $c_{t_2}$ using the terms provided by the assumption. For $c_{t_3,i}$ we take a closer look at the exponent. Applying the definition of $s_1$, $s'$ and $v$, we get

$$
\begin{align*}
-x + \varepsilon_1 &+ \frac{xy[1]}{x[y]} + \varepsilon_2 + \cdots + \frac{xy[n_2]}{x[y]} + \varepsilon_n_2) \end{align*}
\end{equation}
$$

Observe that the first and the third term sum up to $0$ and that the other terms can be simulated using the $g_1$ terms of the assumption.

Also note that if $b = 0$, then $d = e(g_1, g_2) \cdot e(g_1, g_2) = e(g_1, g_2)^\frac{1}{2}(x + \varepsilon_1) = e(g_1, g_2)^\alpha s_1$ is the real KEM key and if $b = 1$, then $d$ is uniformly distributed in $\mathbb{G}_T$.

Simulating $O_{sk}$. On a query $S$ to $O_{sk}$, first compute $w \in \mathbb{Z}_p^{n_2}$ such that $\langle w, M_1^T \rangle = 0$ for all $i \in [n_1]$ such that $\pi(i) \in S$. Note that such a vector $w$ is efficiently computable and we can assume that w.l.o.g. $w[1] = 1$. We implicitly set $r = \delta \sum_{j \in [n_2]} w[j]y[j]+\gamma$, where $\gamma \in \mathbb{Z}_p$. Now we can compute $sk$ as follows:

$$
\begin{align*}
sk_1 &:= g_1^\frac{1}{2} - \frac{\sum_{j \in [n_2]} w[j]y[j]+\gamma}{x[y]} = g_1 - \frac{\sum_{j \in [n_2]} w[j]y[j]}{x[y]}, \\
\delta \sum_{j \in [n_2]} w[j]y[j]+\gamma &:= H(u) \delta \sum_{j \in [n_2]} w[j]y[j]+\gamma \\
\delta \sum_{j \in [n_2]} w[j]y[j]+\gamma &:= g_2^\frac{1}{2} \cdot g_2^\frac{1}{2}
\end{align*}
$$

Observe that $sk_1$ and $sk_3$ can be simulated using the terms provided by the assumption. For $sk_2,u$ we take a closer look at the exponent. Since the value depends on $H(u)$, we distinguish two cases, starting with the case that there exists an index $i$ such that $\pi(i) = u$. Applying the definition of $H(u)$, we get

$$
\begin{align*}
\left( \sum_{j \in [n_2]} M_{i,j}w[j]y[k] \right) + \epsilon_u \left( \sum_{j \in [n_2]} \delta \sum_{j \in [n_2]} w[j]y[j]+\gamma \right) \\
\sum_{j \in [n_2]} M_{i,j}w[j]y[k] + \varepsilon_u \gamma + \sum_{j \in [n_2]} M_{i,j}w[j]y[j]+\gamma + \epsilon_u \gamma \\
\delta \sum_{j \in [n_2]} w[j]y[j]+\gamma + \epsilon_u \gamma \\
\delta \sum_{j \in [n_2]} w[j]y[j]+\gamma + \epsilon_u \gamma \\
\delta \sum_{j \in [n_2]} w[j]y[j]+\gamma + \epsilon_u \gamma
\end{align*}
$$

The first term underneath the curly bracket is $0$ since $w$ is chosen such that $M_iw^T = 0$ for all $\pi(i) \in S$. The other terms can be simulated using the $g_1$ terms of the assumption.

It remains to show that we can simulate $sk_{2,u}$ in case there does not exist an index for all $i$ such that $\pi(i) = u$. In this case $H(u)$ is simply $g_1^\epsilon_u$, so we get

$$
\begin{align*}
\epsilon_u \left( \sum_{j \in [n_2]} w[j]y[j]+\gamma \right) = g_1^\epsilon_u \delta \sum_{j \in [n_2]} w[j]y[j]+\gamma + \epsilon_u \gamma \\
which can be computed using the terms provided by the assumption.

Finalize. At the end, the adversary $A$ will output a bit $b'$ which will also be $D$'s output. Note that if $A$ wins, then $D$ will always output the correct bit except when $T_1$ coincides with $T_0$ which happens with probability $1/p$. This concludes the proof of Theorem 2.
B.2 KP-ABE

In a similar way, we prove security of our KP-ABE scheme (cf. Figure 1) for \( \tau = 1 \) in the random oracle model for parameter \( n = Q_H \), where \( Q_H \) is the number of queries to the random oracle \( H \).

**Theorem 3.** Let \( \lambda \) be the security parameter. For any adversary \( A \) against selective security of KP-ABE with \( \tau = 1 \) that issues at most \( Q_{sk} \) queries to \( O_{sk} \), one query to \( O_{ct} \) and at most \( Q_H \) queries to the random oracle \( H \), there exists an adversary \( D \) such that

\[
\text{Adv}_{\text{KP-ABE},A}^{\text{selective}}(1^\lambda) \leq \text{Adv}_{D,Q_H}^{\text{q-type}}(1^\lambda) + \frac{1}{p}.
\]

**Proof.** The proof is similar to that of Theorem 2. Let \( A \) be an adversary against selective security of the KP-ABE scheme in Figure 1. We construct an adversary \( D \) against our q-type assumption that simulates the security game for \( A \).

\( D \) inputs \((p, G_1, G_2, G_T, e, D^1, D^2, T_b)\) and runs adversary \( A \). First \( A \) commits on a challenge attribute set \( S \). Now \( D \) must simulate the first query to \( O_{mpk} \), the challenge query to \( O_{ct} \) and adaptive queries to \( O_{sk} \) and \( H \).

**Simulating \( O_{mpk} \).** This is done as in the proof of Theorem 2. We implicitly set \( \alpha = \frac{d}{2} \) and output public parameters

\[
\left( p, G_1, G_2, G_T, e, g_1, g_2^{\frac{1}{2}}, e(g_1, g_2^{\frac{1}{2}})^\alpha \right).
\]

**Simulating \( H \).** The random oracle is simulated by lazy sampling as described above. Note that the assumption is parameterized by the maximum number of random oracle queries and we index each element of the vector \( y \) by the values \( s \) queried to \( H \), i.e., \( y[u] \). Thus, on each query which is distinct to previous ones, we will first pick a new exponent \( u \epsilon Z_p \) and output

\[
H(u) := \begin{cases} 
\frac{g_1^{\frac{1}{2}w}}{g_1^s}, & \text{if } \exists u \notin S \\
g_1^s, & \text{otherwise}
\end{cases}
\]

**Simulating \( ct \).** To compute the challenge for \( S \) provided at the beginning, we (implicitly) set \( s = x + \epsilon \) for \( \epsilon \epsilon Z_p \). Then compute and output

\[
ct_{1,u} := H(u)^{x+\epsilon} = g_1^{\epsilon u} \cdot g_1^{\epsilon \epsilon} \\
ct_2 := g_2^{\frac{1}{2}} \cdot g_2^{\frac{1}{2}} \\
d := T_b \cdot e(g_1^{\frac{1}{2}}, g_2^{y[1]})^\epsilon
\]

Note that if \( b = 0 \), then \( d = e(g_1, g_2) \cdot e(g_1, g_2)^{\frac{1}{2}} = e(g_1, g_2)^{\frac{1}{2}(x+\epsilon)} = e(g_1, g_2)^{\alpha s} \) is the real KEM key and if \( b = 1 \), then \( d \) is uniformly distributed in \( G_T \).

**Simulating \( sk \).** On a query \((M, \pi)\) to \( O_{sk} \), first compute \( w \epsilon Z_{n^2}^p \) such that \((w, M^\top) = 0\) for all \( i \in [n_1] \) such that \( \pi(i) \in S \). Note that such a vector \( w \) is efficiently computable and we can assume that w.l.o.g. \( w[1] = 1 \). We implicitly set \( r^\prime = - \delta \sum_{i \in [n_1]} M_i w^\top y[\pi(i)] + \gamma_1 \) and \( v[j] = \frac{w[j]}{\epsilon} + \gamma_j \) for \( j \in [2, n_2] \), where \( \gamma_1, \ldots, \gamma_{n_2} \epsilon Z_p \). Now we can compute \( sk \) as follows:

\[
\text{sk}_1 := (g_2^{\frac{1}{2}})^{-\delta} \sum_{i \in [n_1]} M_i w^\top y[\pi(i)] + \gamma_1 = g_2 \sum_{i \in [n_1]} M_i w^\top y[\pi(i)] \cdot g_2^\gamma \\
\text{sk}_{2,i} := g_1 M_i \frac{w^\top y[\pi(i)]}{\gamma_1} \cdot H(\pi(i))^{-\delta} \sum_{i \in [n_1]} M_i w^\top y[\pi(i)] + \gamma_1
\]

It is easy to see that we can simulate \( sk_1 \) using \( g_2^{\frac{1}{2}} \) and \( g_2^\gamma u \) terms of the assumption. For \( sk_{2,i} \), we take a closer look at the exponent. Since the value depends on \( H(\pi(i)) \), we distinguish two cases, starting with
the case that \( \pi(i) \notin S \). Applying the definition of \( \nu \) and \( H(u) \), we get

\[
M_i \left( \frac{\delta}{x} \frac{\delta w[2]}{x} + \gamma \ldots + \frac{\delta w[n]}{x} \right) - \left( \frac{1}{x^y[H(\pi(i))]} + \epsilon_{\pi(i)} \right) \left( \sum_{k \in [n]} \delta M_k w^T y[\pi(k)] + \gamma \right)
\]

\[
= \frac{\delta M_i w^T}{x} + \sum_{j \in [2,n]} M_{i,j} \gamma_j - \sum_{k \in [n]} \delta M_k w^T y[\pi(k)] \epsilon_{\pi(i)} - \epsilon_{\pi(i)} \gamma
\]

Observe that the first term cancels with the first term underneath the curly bracket and that the other terms can be simulated using the \( g_1 \) terms of the assumption.

It remains to show that we can simulate \( s_{k_2,i} \) for all \( i \) such that \( \pi(i) \in S \). In this case \( H(\pi(i)) \) is simply \( g_1^{\epsilon_{\pi(i)}} \), so we get

\[
\frac{\delta M_i w^T}{x} + \sum_{j \in [2,n]} M_{i,j} \gamma_j - \sum_{k \in [n]} \delta M_k w^T y[\pi(k)] \epsilon_{\pi(i)} - \epsilon_{\pi(i)} \gamma
\]

where the first term is 0 since \( w \) is chosen such that \( M_i w^T = 0 \) for all \( \pi(i) \in S \). The remaining terms can be computed using the terms provided by the assumption.

**Finalize.** At the end, the adversary \( \mathcal{A} \) will output a bit \( \beta' \) which will also be \( D \)'s output. Note that if \( \mathcal{A} \) wins, then \( D \) will always output the correct bit except when \( T_1 \) coincides with \( T_0 \) which happens with probability \( 1/p \). This concludes the proof of Theorem 3.

### B.3 Relation to Other \( q \)-type Assumptions

In order to justify our new \( q \)-type assumption, we show that it is implied by another known assumption. We recall the expanded Diffie-Hellman exponent assumption-2 (EDHE2) introduced by Attrapadung [7]. The assumption is defined for a group \( G_1 \) of prime order \( p \) with generator \( g_1 \). It is parameterized by two integers \( n, m \) and says that given

\[
\forall i \in [n], j, j' \in [m], j \neq j' : \quad g_1^{a_i}, g_1^{a_i^2}, g_1^{a_i^3}, g_1^{a_i^4} \text{ are random}
\]

\[
\forall i \in [0, n-1] : \quad g_1^{a_i^2}, g_1^{a_i^3}, g_1^{a_i^4}, g_1^{a_i^{4_1}}, g_1^{a_i^{4_2}}, g_1^{a_i^{4_3}}, g_1^{a_i^{4_4}}, g_1^{a_i^{4_5}}, g_1^{a_i^{4_6}}
\]

\[
\forall i \in [2n-1], j, j' \in [m], j \neq j' : \quad g_1^{a_i^{4_7}}, g_1^{a_i^{4_8}}, g_1^{a_i^{4_9}}, g_1^{a_i^{4_{10}}}, g_1^{a_i^{4_{11}}}, g_1^{a_i^{4_{12}}}, g_1^{a_i^{4_{13}}}, g_1^{a_i^{4_{14}}}, g_1^{a_i^{4_{15}}}
\]

\[
\forall i \in [2n-1], i \neq n, j \in [m] : \quad g_1^{a_i^{4_{16}}}, g_1^{a_i^{4_{17}}}, g_1^{a_i^{4_{18}}}, g_1^{a_i^{4_{19}}}, g_1^{a_i^{4_{20}}}, g_1^{a_i^{4_{21}}}, g_1^{a_i^{4_{22}}}, g_1^{a_i^{4_{23}}}, g_1^{a_i^{4_{24}}}
\]

\[
\forall i \in [2n-1], j, j' \in [m] : \quad g_1^{a_i^{4_{25}}}, g_1^{a_i^{4_{26}}}, g_1^{a_i^{4_{27}}}, g_1^{a_i^{4_{28}}}, g_1^{a_i^{4_{29}}}, g_1^{a_i^{4_{30}}}, g_1^{a_i^{4_{31}}}, g_1^{a_i^{4_{32}}}, g_1^{a_i^{4_{33}}}
\]

\[
\tilde{T}_0 = g_1^{ab_2} \text{ is computationally indistinguishable from a random element } \tilde{T}_0 \in G_1.
\]

We will look at the asymmetric version of the assumption where the adversary is also given \( g_2 \) raised to the same exponents and the challenge is provided in \( G_2 \). The terms highlighted in gray will be needed for our reduction. Then we can show that the asymmetric \((2,n)\)-EDHE2 assumption implies our \( n\)-\( q \)-type assumption. Given an adversary \( \mathcal{A} \) against the \( n\)-\( q \)-type assumption, we construct a reduction \( \mathcal{B} \) as follows. \( \mathcal{B} \) inputs the terms of the asymmetric \((2,n)\)-EDHE2 assumption as described above, it then
(implicitly) sets
\[ g_1 = g_1^{ac}, \]
\[ g_2 = g_2^{ab}, \]
\[ \delta = \frac{b\gamma}{c} \text{ where } \gamma \in \mathbb{Z}_p, \]
\[ x = \frac{\omega}{u} \text{ where } \omega \in \mathbb{Z}_p, \]
\[ y[i] = \frac{d_i}{ab} \]

It computes the terms of the \( n \)-q-type assumption as
\[ \tilde{g}_1^i = (\tilde{g}_1^{ac})^{\tilde{\pi}} = (\tilde{g}_1^{c})^{\tilde{\omega}} \]
\[ \tilde{g}_1^{y[i]} = (\tilde{g}_1^{ac})^{\tilde{\pi} \cdot \tilde{d}_i} = (\tilde{g}_1^{\frac{c\tilde{\delta}}{\tilde{\mu}_p}})^{\tilde{\delta}} \]
\[ \tilde{g}_2^i = (\tilde{g}_2^{ab})^{\tilde{\pi}} = (\tilde{g}_2^{ac})^{\tilde{\gamma}} \]
\[ \tilde{g}_2^{y[i]} = (\tilde{g}_2^{ab})^{\tilde{\pi} \cdot \tilde{d}_i} = \tilde{g}_2^{d_i} \]
\[ T_b = e(g_1^{\frac{c\tilde{\delta}}{\tilde{\mu}_p}}, T_b) \]

Note that for \( T_0 = g_2^{abz} \), we have \( T_b = e(g_1^{\frac{c\tilde{\delta}}{\tilde{\mu}_p}}, g_2^{abz}) = e(g_1^{ac}, g_2^{ab}) = e(g_1, g_2) \) and for \( T_1 \in \mathbb{Z}_p \), \( T_b \) is a random group element in \( G_T \). Thus, if \( A \) outputs the correct bit, so does the reduction \( B \).

C ABGW Schemes

Below are the CP-ABE and KP-ABE scheme of ABGW [6] that we implemented. \( M_1, \pi \) are defined as in Section 2.2.

The CP-ABE scheme is defined as follows.

- **Setup**(1\(^\lambda\)). Run **GroupGen**(1\(^\lambda\)) to obtain \( G := (p, G_1, G_2, G_T, e, g_1, g_2) \). Pick \((\alpha || b) \in \mathbb{Z}_p \times \mathbb{Z}_p^4\). Output \((G, (g_1^{[b][i]}), e(g_1, g_2)^a)\) as the master public key \( \text{mpk} \) and \((\alpha, b)\) as the master secret key \( \text{msk} \).

- **KeyGen**\((\text{msk}, S)\). Pick \( r \in \mathbb{Z}_p \). Compute

\[ \text{sk}_1 := g_2^{a-b[4]r}, \quad \text{sk}_2 := g_2^r \]

as well as

\[ \text{sk}_{3,u} := g_2^{b[u][i]}, \quad \text{sk}_{3,u} := g_2^{b[u][i]}, \]

for all \( u \in S \). Output the secret key \( \text{sk} := (\text{sk}_1, \text{sk}_2, (\text{sk}_{3,u})_{u \in S}) \).

- **Enc**(\( \text{mpk}, (M_1, \pi) \)). Pick \((s_1 || s') || v) \in \mathbb{Z}_p \times \mathbb{Z}_p^{n_1} \times \mathbb{Z}_p^{n_2} \). Let \( \mu_i := M_1(s_1 || v)^T \) for all \( i \in [n_1] \). Then compute

\[ \text{ct}_{1,i} := g_1^{\mu_i}, \quad \text{ct}_{2,i} := g_1^{-b[3][s'[i]]}, \quad g_1^{b[i][\mu_i]}, \quad \text{ct}_{3,i} := \left( g_1^{[b][i]}, g_1^{b[2][\pi(i)]} \right)^{s'[i]} \]

and \( d := e(g_1, g_2)^{\alpha s_1} \). Output the ciphertext \( \text{ct} := (\text{ct}_{1,i}, \text{ct}_{2,i}, \text{ct}_{3,i})_{i \in [n_1]} \) and key \( d \).
The KP-ABE scheme is defined as follows.

- **Setup**(1^λ). Run **GroupGen**(1^λ) to obtain \( G := (p, G_1, G_2, G_T, e, g_1, g_2) \). Pick \( (\alpha, b) \leftarrow \mathbb{Z}_p \times \mathbb{Z}_p \). Output \( (G, g_b^{[1]}, g_b^{[2]}, e(g_1, g_2)^\alpha) \) as the master public key mpk and \( (\alpha, b) \) as the master secret key msk.

- **KeyGen**(msk, (M, π)). Pick \( r \leftarrow \mathbb{Z}_p \). Let \( \mu_i := M_i(\alpha \parallel r) \) for all \( i \in [n_1] \). Then compute

\[
\begin{align*}
\text{sk}_{1,i} := g_{2^{b_i}}, \quad \text{sk}_{2,i} := g_{2^{g_1^{b_1}[u] \cdot g_1^{b_2}[u]}}^{\mu_i}
\end{align*}
\]

and output the secret key sk := (sk_{1,i}, sk_{2,i})_{i \in [n_1]}.

- **Enc**(mpk, S). Pick \( s \leftarrow \mathbb{Z}_p \times \mathbb{Z}_p^{|S|} \). For all \( u \in S \), compute

\[
\begin{align*}
\text{ct}_{1,u} := g_1^{s_1 - s[u]}, \quad \text{ct}_{2,u} := \left(g_1^{b_1}, g_1^{b_2} \right)^{s[u]}
\end{align*}
\]

and \( d := e(g_1, g_2)^{\alpha s_1} \). Output the ciphertext ct := (ct_{1,u}, ct_{2,u})_{u \in S} and key d.

- **Dec**(mpk, S, (M, π), ct, sk). If S satisfies (M, π), there exist constants \( \{\gamma_i\}_{i \in I} \) s.t. \( \sum_{i \in I} \gamma_iM_i = (1, 0, \ldots, 0) \). Reconstruct d by computing

\[
\prod_{i \in I} e(\text{ct}_{1,\pi(i)}^{\gamma_i}; \text{sk}_{1,i})^{\gamma_i} \prod_{i \in I} e(\text{ct}_{2,\pi(i)}^{\gamma_i}; \text{sk}_{2,i})^{\gamma_i}
\]

and output the result.