Semi-adaptive Attribute-Based Encryption and Improved Delegation for Boolean Formula*

Jie Chen[1,**] and Hoeteck Wee[2,***]

[1] Department of Computer Science and Technology, East China Normal University
s080001@e.ntu.edu.sg
[2] École Normale Supérieure, Paris
wee@di.ens.fr

Abstract. We consider *semi-adaptive* security for attribute-based encryption, where the adversary specifies the challenge attribute vector after it sees the public parameters but before it makes any secret key queries. We present two constructions of semi-adaptive attribute-based encryption under static assumptions with *short* ciphertexts. Previous constructions with short ciphertexts either achieve the weaker notion of selective security, or require parameterized assumptions.

As an application, we obtain improved delegation schemes for Boolean formula with *semi-adaptive* soundness, where correctness of the computation is guaranteed even if the client's input is chosen adaptively depending on its public key. Previous delegation schemes for formula achieve one of adaptive soundness, constant communication complexity, or security under static assumptions; we show how to achieve semi-adaptive soundness and the last two simultaneously.

1 Introduction

Attribute-based encryption (ABE) [33, 20] is an emerging paradigm for public-key encryption which enables fine-grained control of access to encrypted data. In traditional public-key encryption, access to the encrypted data is all or nothing: given the secret key, one can decrypt and read the entire plaintext, but without it, nothing about the plaintext is revealed (other than its length). In ABE, a

* The research leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 339563 CryptoCloud). A longer version of this work appears in [11].

** Supported by Science and Technology Commission of Shanghai Municipality under Grants 14YF1404200, 13JC1403500, and the National Natural Science Foundation of China Grant No. 61172085. Part of this work was done at Nanyang Technological University, supported by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03.

*** CNRS (UMR 8548) and INRIA. Supported in part by the French ANR-12-INSE-0014 SIMPATIC Project. Part of this work was done at Columbia University, supported by NSF Award CNS-1319021, and at Ruhr-Universität Bochum as a Research Fellow of the Alexander von Humboldt Foundation.

M. Abdalla and R. De Prisco (Eds.): SCN 2014, LNCS 8642, pp. 277–297, 2014.

ciphertext is labeled with an attribute vector $\mathbf{x}$, and a secret key is associated with an access policy specified as a Boolean formula, and the secret key decrypts the ciphertext if and only if $\mathbf{x}$ satisfies the access policy.[1] It is easy to see that ABE is a generalization of identity-based encryption (IBE) [34, 5, 14]. The security requirement for ABE stipulates that it resists collusion attacks, namely any group of users collectively learns nothing about the plaintext if none of them is individually authorized to decrypt the ciphertext.

Delegation. A delegation scheme allows a computationally weak client to delegate expensive computations to the cloud, with the assurance that a malicious cloud cannot convince the client to accept an incorrect computation [19, 17, 4, 15]. Recent work of Parno, Raykova and Vaikuntanathan [32] showed that any ABE with encryption time at most linear in the length of the attribute vector immediately yields a delegation scheme for Boolean formula. There is an initial pre-processing phase which fixes the formula f the client wishes to compute and produces some public key. Afterwards, to delegate computation on an input x, the client only needs to send a single message. Moreover, the ensuing delegation scheme satisfies public delegatability, namely anyone can delegate computations to the cloud; as well as public verifiability, namely anyone can check the cloud's work (given a "verification" key published by the client).

State of the Art. Since the introduction of ABE and motivated in part by the connection to delegation, there is now a large body of work providing constructions with incomparable trade-offs amongst efficiency, security guarantees and security assumptions [20, 2, 27, 31, 26]; a summary of this work is presented in Fig 1. A key measure of efficiency is the ciphertext size and the encryption time; ideally, we want this to depend at most linearly in the length of the attribute vector and independent of the size of the access structure. For security guarantees, the two primary notions are selective and adaptive security; in the more restrictive setting of selective security, the adversary must specify the challenge attribute vector prior to seeing the public parameters. Finally, the security of the schemes rely on the assumed hardness of some computational problem in bilinear groups; here, we prefer prime-order instantiations over composite-order ones, and static assumptions over parameterized ones.

1.1 Our Contributions

We introduce the notion of *semi-adaptive* security for ABE and delegation. In ABE, this means that the adversary specifies the challenge attribute vector after it sees the public parameters but before it makes any secret key queries. This

[1] This is typically referred to as key-policy ABE in the literature, which is the focus of this paper. A different line of works, e.g. [13, 21, 37, 27, 26], considers ciphertext-policy ABE, where the ciphertext is labeled with a formula and the secret key is associated with an attribute vector.

reference	security	Enc time	CT size	MPK size	SK size	group	assumption
GPSW06 [20]	selective	$O(n)^*$	$O(n)^*$	$O(n)$	$O(\ell)$	prime	static
ALP11 [2]		$O(n)$	$O(1)$	$O(n)$	$O(n\ell)$	prime	non-static
ALP11+LW10		$O(n)$	$O(1)$	$O(n)$	$O(n\ell)$	composite	static
T14 [35]		$O(n)$	$O(1)$	$O(n)$	$O(n\ell)$	prime	static
LOSTW10 [27]	adaptive	$O(nM)^*$	$O(nM)^*$	$O(nM)$	$O(\ell)$	composite	static
OT10 [31]		$O(nM)^*$	$O(nM)^*$	$O(nM)$	$O(\ell)$	prime	static
LW12 [26]		$O(n)^*$	$O(n)^*$	$O(n)$	$O(\ell)$	prime	non-static
A14 [1]		$O(n)$	$O(1)$	$O(n)$	$O(n\ell)$	composite	non-static
Construction 1	semi-	$O(n)$	$O(1)$	$O(n)$	$O(n\ell)$	composite	static
Construction 2	adaptive	$O(n)^*$	$O(n)^*$	$O(n)$	$O(\ell)$	prime	static

Fig. 1. Summary of existing KP-ABE schemes. Here, n denotes the universe size, M is the maximum number of times an attribute may be used, and $\ell \leq nM$ is the number of rows in the matrix $\mathbf{M}$ of the access structure. Encryption time is given in terms of group operations, and CT, PP, SK sizes are given in terms of group elements. For CT, we omit the additive overhead of n *bits* in order to transmit the attribute vector. For the quantities marked with *, n may be replaced with number of non-zero entries in the attribute vector $\mathbf{x} \in \{0,1\}^n$, which could be much smaller than n. Note that ALP11, T14 and A14 achieve large universe, we restrict the attribute universe to $[n]$ for comparison.

is stronger than selective security but weaker than adaptive security. In delegation, this means that the client's input may depend on the public key but is independent of the worker's evaluation key. In addition, we provide new constructions of efficient semi-adaptively secure ABE and delegation schemes under static assumptions.

New ABE Schemes. Our first result is a semi-adaptively secure ABE whose efficiency matches the state-of-the-art selectively secure ABE [2]:

> **(Informal Theorem)** There exists a semi-adaptively secure ABE with constant-size ciphertexts. Encryption time is linear in the length of the attribute vector and independent of the size of the access structure. The security of the scheme is based on static assumptions in composite-order groups.

We also achieve an analogous result in prime-order groups based on the SXDH Assumption; however, the ciphertext size is linear in the length of the attribute vector. Throughout this work, when we refer to ciphertext size, we measure the number of group elements, and we omit the additive overhead of n *bits* needed to transmit the attribute vector.

New Delegation Schemes. Starting from our semi-adaptively secure ABE, we obtain improved delegation schemes for Boolean formula with *semi-adaptive* soundness, where correctness of the computation is guaranteed even if the client's input is chosen adaptively depending on its public key. We note that achieving semi-adaptive soundness is important in practice, since we would like to reuse the

reference	security	$\lvert\mathsf{EK}_\mathsf{F}\rvert$	client's communication in bits	worker's complexity	groups	assumptions
GPSW06 [20]	selective	$O(\ell)$	$O(n\lambda)$	$O(\ell)$	prime	static
ALP11 [2]		$O(n\ell)$	$n + O(\lambda)$	$O(n\ell)$	prime	non-static
T14 [35]		$O(n\ell)$	$n + O(\lambda)$	$O(n\ell)$	prime	static
GGPR13 [18]	adaptive	$O(\ell)$	$n + O(\lambda)$	$O(\ell)$	prime	non-static
LW12 [26]		$O(\ell)$	$O(n\lambda)$	$O(\ell)$	prime	non-static
A14 [1]		$O(n\ell)$	$n + O(\lambda)$	$O(n\ell)$	composite	non-static
Construction 1	semi-adaptive	$O(n\ell)$	$n + O(\lambda)$	$O(n\ell)$	composite	static
Construction 2		$O(\ell)$	$O(n\lambda)$	$O(\ell)$	prime	static

Fig. 2. Summary of existing publicly verifiable computation schemes. GGPR13 supports NC. The remaining schemes only support NC^1 and are obtained using the transformation of [32]. Here, $\lvert\mathsf{EK}_\mathsf{F}\rvert$ is the worker's evaluation key, n is the bit length of the input and ℓ is the size of the formula. In all the schemes, the public key is $O(n)$ group elements, delegation and verification complexity of client is $O(n)$ group operations, computation complexity of worker is also given in terms of group operations.

same public key across multiple inputs, which could lead to correlation between the input and the public key. Previous delegation schemes for formula achieve one of adaptive soundness [26, 18], constant communication complexity[2] [2], or security under static assumptions [20]; we achieve semi-adaptive soundness and the last two simultaneously. We compare our schemes with prior works in Fig 2. We stress that in applications such as delegating computation from mobile devices on cellular networks where bandwidth is a premium, reducing the client's communication from $O(n\lambda)$ bits to $n + O(\lambda)$ bits represents substantial savings.

1.2 Our Techniques

Following our recent works [38, 9] and inspired in part by [26], we rely on Waters' dual system encryption methodology [36, 25] to reduce the problem of building a (public-key) semi-adaptively secure ABE to that of building a private-key selectively secure ABE. Recall that dual system encryption is typically implemented by designing a "semi-functional space" where semi-functional components of keys and ciphertexts will behave like a parallel copy of the normal components of the system, except divorced from the public parameters. In particular, we will embed the private-key selectively secure ABE into the semi-functional space.

We proceed to outline the constructions of private-key ABE with short ciphertexts:

– For our composite-order scheme with constant-size ciphertext, we use a private-key variant of the selectively secure ABE scheme of Attrapadung, Libert and Panafieu (ALP) in [2]. Our main insight is that in the private-key

[2] Here, we refer to the client's communication overhead beyond sending the n-bit input, as measured in group elements.

setting with a single challenge ciphertext, we can replace the use of parameterized assumptions in the ALP scheme with the basic DDH assumption. Roughly speaking, fix an attribute i that does not appear in the challenge attribute. We can then rely on the DDH assumption to mask all the LSSS shares of the master secret key corresponding to attribute i (c.f. Section 3 overview and Lemma 2).[3] The formal security proof is more involved since we need to instantiate this argument within the dual system framework.

– For our prime-order scheme with $O(n)$-size ciphertext, the private-key selectively secure ABE we use is essentially that of Goyal et al. [20], which is in fact a public-key scheme and yields ciphertexts of length $O(n)$. To combine this scheme with the dual system framework, we rely on dual pairing vector spaces [29, 30, 16, 24, 12]. Here, we will also use the SXDH assumption to boost *statistical* entropy in the semi-functional key space into arbitrarily large amounts of *computational* entropy in the same space as we will need to mask an arbitrarily large number of shares corresponding to a single attribute.

For both schemes, we are able to exploit random self-reducibility to obtain security loss that do not depend on the number of secret key queries or the size of the boolean formula (but may depend on the input size n). In contrast, all known adaptively secure ABE schemes incur a loss that is at least linear in both the number of secret key queries and the size of the boolean formula (sometimes implicitly, by either making a "one-use" restriction or using a parameterized assumption).

Additional Related Work. In an independent work, Takashima [35] proposed a selectively secure KP-ABE scheme with constant-size ciphertexts under the DLIN assumption, which results in a delegation scheme with constant communication complexity and security under static assumptions but only achieving selective soundness. Upon learning of our work, Takashima showed that his scheme also achieves semi-adaptive security, thereby resolving a natural open problem from this work. Gennaro, Gentry, Parno and Raykova [18] constructed a delegation scheme achieving adaptive soundness and supporting NC but its security relies on parameterized assumptions.

Organization. We present our composite-order construction in Section 3. We provide our prime-order construction, the delegation schemes and associated definitions in the full version of this paper [11].

[3] In an earlier submission, an anonymous reviewer asked if it is possible to obtain the composite-order scheme by combining the Lewko-Waters ABE [26] with the ALP scheme. We clarify here that this approach (should it pan out) would inherit the parameterized assumption from [2]. In particular, none of the prior works either implicitly or explicitly build a private-key ABE with constant-size ciphertexts from static assumptions.

2 Preliminaries

Notation. We denote by $s \leftarrow_{\mathrm{R}} S$ the fact that s is picked uniformly at random from a finite set S and by $x, y, z \leftarrow_{\mathrm{R}} S$ that all x, y, z are picked independently and uniformly at random from S. By PPT, we denote a probabilistic polynomial-time algorithm. Throughout, we use 1^λ as the security parameter. We use $\cdot$ to denote multiplication (or group operation) as well as component-wise multiplication. We use lower case boldface to denote (column) vectors over scalars and upper case boldface to denote vectors of group elements as well as matrices. Given two vectors $\mathbf{x} = (x_1, x_2, \ldots), \mathbf{y} = (y_1, y_2, \ldots)$ over scalars, we use $\langle \mathbf{x}, \mathbf{y} \rangle$ to denote the standard dot product $\mathbf{x}^\top \mathbf{y}$. Given a group element g, we write $g^{\mathbf{x}}$ to denote $(g^{x_1}, g^{x_2}, \ldots)$; we define $g^{\mathbf{A}}$ where $\mathbf{A}$ is a matrix in an analogous way.

2.1 Access Structures

We define (monotone) access structures using the language of (monotone) span programs [22].

Definition 1 (access structure [3, 22]). *A* (monotone) access structure $\mathbb{A}$ *for attribute universe* $[n]$ *is a pair* $(\mathbf{M}, \rho)$ *where* $\mathbf{M}$ *is a* $\ell \times \ell'$ *matrix over* $\mathbb{Z}_N$ *and* $\rho : [\ell] \to [n]$. *Given* $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$, *we say that*

$$\mathbf{x} \text{ satisfies } \mathbb{A} \text{ iff } \mathbf{1} \in \mathsf{span}\langle \mathbf{M}_{\mathbf{x}} \rangle.$$

Here, $\mathbf{1} := (1, 0, \ldots, 0) \in \mathbb{Z}_N^{\ell'}$ *is a row vector;* $\mathbf{M}_{\mathbf{x}}$ *denotes the collection of vectors* $\{\mathbf{M}_j : x_{\rho(j)} = 1\}$ *where* $\mathbf{M}_j$ *denotes the* j*'th row of* $\mathbf{M}$*; and* span *refers to linear span of collection of (row) vectors over* $\mathbb{Z}_N$.

That is, $\mathbf{x}$ satisfies $\mathbb{A}$ iff there exists constants $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_N$ such that

$$\sum_{j : x_{\rho(j)} = 1} \omega_j \mathbf{M}_j = \mathbf{1}.$$

Observe that the constants $\{\omega_j\}$ can be computed in time polynomial in the size of the matrix $\mathbf{M}$ via Gaussian elimination.

2.2 Key-Policy Attribute-Based Encryption

A KP-ABE scheme consists of four algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$:

- $\mathsf{Setup}(1^\lambda, [n]) \to (\mathrm{MPK}, \mathrm{MSK})$. The setup algorithm takes in a security parameter 1^λ, and an attribute universe $[n]$. It outputs public parameters MPK and a master secret key MSK.
- $\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}, m) \to \mathrm{CT}_{\mathbf{x}}$. The encryption algorithm takes in MPK, an attribute vector $\mathbf{x}$, and a message m. It outputs a ciphertext $\mathrm{CT}_{\mathbf{x}}$.
- $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbb{A}) \to \mathrm{SK}_{\mathbb{A}}$. The key generation algorithm takes in MPK, MSK, and an access structure $\mathbb{A} := (\mathbf{M}, \rho)$. It outputs a secret key $\mathrm{SK}_{\mathbb{A}}$.
- $\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbb{A}}, \mathrm{CT}_{\mathbf{x}}) \to m$. The decryption algorithm takes in MPK, a secret key $\mathrm{SK}_{\mathbb{A}}$ for an access structure $\mathbb{A}$, and a ciphertext $\mathrm{CT}_{\mathbf{x}}$ encrypted under an attribute vector $\mathbf{x}$. It outputs a message m if $\mathbf{x}$ satisfies $\mathbb{A}$.

Correctness. For all $(\text{MPK}, \text{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, [n])$, all access structures $\mathbb{A}$, all decryption keys $\text{SK}_\mathbb{A}$, all messages m, all $\mathbf{x}$ satisfying $\mathbb{A}$, we have $\Pr[\mathsf{Dec}(\text{MPK}, \text{SK}_\mathbb{A}, \mathsf{Enc}(\text{MPK}, \mathbf{x}, m)) = m] = 1$.

2.3 Semi-adaptive Security Model

We now formalize the notation of *semi-adaptive* security for KP-ABE. Briefly, the adversary specifies the challenge attribute vector after it sees the public parameters and before it makes any secret key queries. The security game is defined by the following experiment, played by a challenger and an adversary $\mathcal{A}$.

Setup. The challenger runs the setup algorithm to generate (MPK, MSK). It gives MPK to $\mathcal{A}$.

Challenge Attribute. $\mathcal{A}$ gives the challenger a challenge $\mathbf{x}^*$.

Phase 1. $\mathcal{A}$ adaptively requests keys for access structures $\mathbb{A}$ with the constraint $\mathbf{x}^*$ does not satisfy $\mathbb{A}$. The challenger responds with the corresponding secret key $\mathsf{SK}_\mathbb{A}$, which it generates by running the key generation algorithm.

Challenge Ciphertext. $\mathcal{A}$ submits two equal-length messages m_0 and m_1. The challenger picks $\beta \leftarrow_\text{R} \{0, 1\}$, and encrypts m_β under $\mathbf{x}^*$ by running the encryption algorithm. It sends the ciphertext to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues to issue key queries as in **Phase 1**.

Guess. $\mathcal{A}$ must output a guess β' for β.

The advantage $\mathsf{Adv}_\mathcal{A}^{\text{KP-ABE}}(\lambda)$ of an adversary $\mathcal{A}$ is defined to be $\Pr[\beta' = \beta] - 1/2$.

Definition 2. *A KP-ABE scheme is* semi-adaptively secure *if all PPT adversaries achieve at most a negligible advantage in the above security game.*

2.4 Composite Order Bilinear Groups

Composite order bilinear groups were first introduced in [7] and used in [23, 25, 27]. A generator $\mathcal{G}$ takes as input a security parameter 1^λ and outputs a description $\mathbb{G} := (N, G_N, G_T, e)$, where N is product of distinct primes of $\Theta(\lambda)$ bits, G_N and G_T are cyclic groups of order N, and $e : G_N \times G_N \rightarrow G_T$ is a map with the following properties:

1. (Bilinearity) $\forall g, h \in G_N, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$;
2. (Non-degeneracy) $\exists g \in G_N$ such that $e(g, g)$ has order N in G_T.

We require that the group operations in G_N and G_T as well the bilinear map e are computable in deterministic polynomial time with respect to λ. Furthermore, the group descriptions of G_N and G_T include generators of the respective cyclic groups. We use G_n to denote the subgroup of G_N of order n, where n divides N.

Computational Assumptions. We now state the three static assumptions that are required in our security proof. The first two assumptions are introduced in [25] and also used in [27]. The third assumption which basically asserts that the DDH problem is hard in the G_{p_2}-subgroup. This assumption is essentially implied by the composite 3-party Diffie-Hellman (3PDH) assumption in [6]. We provide more discussion and justification of this assumption in the full version of this paper [11]. All three assumptions hold in the generic group model under the assumption finding a non-trivial factor of N is hard.

Assumption 1. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned}
&\mathbb{G} := (N = p_1p_2p_3, G_N, G_T, e) \leftarrow_{\mathrm{R}} \mathcal{G},\\
&g_1, U_1 \leftarrow_{\mathrm{R}} G_{p_1},\ U_2 \leftarrow_{\mathrm{R}} G_{p_2},\ g_3 \leftarrow_{\mathrm{R}} G_{p_3},\\
&T_0 \leftarrow_{\mathrm{R}} G_{p_1},\ T_1 \leftarrow_{\mathrm{R}} G_{p_1p_2},\\
&D := (\mathbb{G}; g_1, U_1U_2, g_3).
\end{aligned}$$

We assume that for any PPT algorithm $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{AS1}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]\right|$$

is negligible in the security parameter λ.

Assumption 2. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned}
&\mathbb{G} := (N = p_1p_2p_3, G_N, G_T, e) \leftarrow_{\mathrm{R}} \mathcal{G},\\
&\alpha, s \leftarrow_{\mathrm{R}} \mathbb{Z}_N,\\
&g_1 \leftarrow_{\mathrm{R}} G_{p_1},\ g_2, X_2, Y_2 \leftarrow_{\mathrm{R}} G_{p_2},\ g_3 \leftarrow_{\mathrm{R}} G_{p_3},\\
&T_0 := e(g_1, g_1)^{\alpha s},\ T_1 \leftarrow_{\mathrm{R}} G_T,\\
&D := (\mathbb{G}; g_1, g_1^{\alpha}X_2, g_1^{s}Y_2, g_2, g_3).
\end{aligned}$$

We assume that for any PPT algorithm $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{AS2}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]\right|$$

is negligible in the security parameter λ.

Assumption 3. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned}
&\mathbb{G} := (N = p_1p_2p_3, G_N, G_T, e) \leftarrow_{\mathrm{R}} \mathcal{G},\\
&x, y, z \leftarrow_{\mathrm{R}} \mathbb{Z}_N,\\
&g_1, U_1 \leftarrow_{\mathrm{R}} G_{p_1},\ g_2, U_2 \leftarrow_{\mathrm{R}} G_{p_2},\ g_3, X_3, Y_3, U_3, W_3 \leftarrow_{\mathrm{R}} G_{p_3},\\
&T_0 := g_2^{xy}W_3,\ T_1 := g_2^{xy+z}W_3,\\
&D := (\mathbb{G}; g_1, U_1U_2, g_2^{x}X_3, g_2^{y}Y_3, g_2U_3, g_3).
\end{aligned}$$

We assume that for any PPT algorithm $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{AS3}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]\right|$$

is negligible in the security parameter λ.

3 Semi-adaptive ABE with Constant-Size Ciphertext

Overview. The starting point of our construction is the following variant of the ALP KP-ABE in [2]:

$$\begin{aligned}\text{MPK} &:= (g, g^{\mathbf{w}}, e(g,g)^{\alpha})\\ \text{CT}_{\mathbf{x}} &:= (g^{s}, g^{s\langle \mathbf{w},\mathbf{x}\rangle}, e(g,g)^{\alpha s}\cdot m)\\ \text{SK}_{\mathbb{A}} &:= (g^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}}, g^{r_j} \ : \ j \in [\ell])\end{aligned}$$

where $\alpha_1, \ldots, \alpha_\ell$ are LSSS shares of α for the access structure $\mathbb{A}$. Our construction proceeds by embedding this scheme into composite-order groups. As noted in the introduction, our main insight is to analyze this scheme in the private-key, selective setting. Fix a selective challenge $\mathbf{x}^* \in \{0,1\}^n$ and an index $k \in [n]$ and an access structure $\mathbb{A}$ not satisfied by $\mathbf{x}^*$. We proceed via a case analysis to argue that $\text{SK}_{\mathbb{A}}$ hides α computationally:

- if $x^*_k = 0$, then the shares $\{\alpha_j : \rho(j) = k\}$ reveal no information about α via the secret sharing property.
- if $x^*_k = 1$, then the ciphertext reveals no information about w_k (and since we are in the private-key setting, there is no MPK). Then, by the DDH assumption, $\{g^{\alpha_j + r_j w_k}, g^{r_j} : \rho(j) = k\}$ computationally hides α_j.

The formal security proof is more involved since we need to instantiate this argument within the dual system framework.

3.1 Construction

- $\mathsf{Setup}(1^\lambda, [n])$: On input an attribute universe $[n]$, generate $\mathbb{G} := (N = p_1 p_2 p_3, G_N, G_T, e) \leftarrow_{\text{R}} \mathcal{G}$, pick $\alpha \leftarrow_{\text{R}} \mathbb{Z}_N, \mathbf{w} \leftarrow_{\text{R}} \mathbb{Z}_N^n$ and output

$$\text{MPK} := (\ \mathbb{G}, e(g_1,g_1)^{\alpha}, g_1, g_1^{\mathbf{w}}\ ) \quad \text{and} \quad \text{MSK} := (\ \alpha, \mathbf{w}, g_2, g_3\ ).$$

- $\mathsf{Enc}(\text{MPK}, \mathbf{x}, m)$: On input an attribute vector $\mathbf{x} := (x_1, \ldots, x_n) \in \{0,1\}^n$ and $m \in G_T$, output

$$\text{CT}_{\mathbf{x}} := \left(\ C_0 := g_1^{s},\ C_1 := g_1^{s\langle \mathbf{w},\mathbf{x}\rangle},\ C_2 := e(g_1,g_1)^{\alpha s}\cdot m\ \right),$$

  where $s \leftarrow_{\text{R}} \mathbb{Z}_N$.

- $\mathsf{KeyGen}(\text{MPK}, \text{MSK}, \mathbb{A} := (\mathbf{M}, \rho))$: On input an access structure $\mathbb{A} := (\mathbf{M}, \rho)$, where $\mathbf{M} \in \mathbb{Z}_N^{\ell \times \ell'}$ and $\rho : [\ell] \to [n]$, pick a random vector $\mathbf{u} \leftarrow_{\text{R}} \mathbb{Z}_N^{\ell'}$ such that $\mathbf{1u} = \alpha$ and set $\alpha_j := \mathbf{M}_j \mathbf{u}$, $j \in [\ell]$.[4] Output

$$\text{SK}_{\mathbb{A}} := \left(\ \boldsymbol{D}_j := g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j,\ D_{0,j} := g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j \ : \ j \in [\ell]\ \right),$$

  where $r_1, r'_1, \ldots, r_\ell, r'_\ell \leftarrow_{\text{R}} \mathbb{Z}_N$; $\boldsymbol{X}_j \leftarrow_{\text{R}} G_{p_3}^n$; $Z_j \leftarrow_{\text{R}} G_{p_3}$, and $(\mathbf{e}_1, \ldots, \mathbf{e}_n)$ is the standard basis for $\mathbb{Z}_N^n$.

[4] The α_j's do in fact correspond to LSSS secret shares of α, distributed across n parties, where the i'th party receive $|\rho^{-1}(i)|$ shares, given by $\{\alpha_j : \rho(j) = i\}$.

– $\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbb{A}}, \mathrm{CT}_{\mathbf{x}})$: If $\mathbf{x}$ satisfies $\mathbb{A}$, compute $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_N$ such that

$$\sum_{j:x_{\rho(j)}=1} \omega_j \mathbf{M}_j = \mathbf{1}.$$

Then, compute[5]

$$e(g_1, g_1)^{\alpha s} \leftarrow \prod_{j:x_{\rho(j)}=1} \Big(e(C_0^{\mathbf{x}}, \boldsymbol{D}_j) \cdot e(C_1, D_{0,j})^{-1} \Big)^{\omega_j},$$

and recover the message as $m \leftarrow C_2 / e(g_1, g_1)^{\alpha s} \in G_T$.

Correctness. Observe that

$$\begin{aligned}
& e(C_0^{\mathbf{x}}, \boldsymbol{D}_j) \cdot e(C_1, D_{0,j})^{-1} \\
&= e((g_1^s)^{\mathbf{x}}, g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j) \cdot e(g_1^{s\langle \mathbf{w}, \mathbf{x} \rangle}, g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j)^{-1} \\
&= e(g_1, g_1)^{\alpha_j s \langle \mathbf{e}_{\rho(j)}, \mathbf{x} \rangle} \cdot e(g_1, g_1)^{r_j s \langle \mathbf{w}, \mathbf{x} \rangle} \cdot e(g_1, g_1)^{-r_j s \langle \mathbf{w}, \mathbf{x} \rangle} \\
&= e(g_1, g_1)^{\alpha_j s}.
\end{aligned}$$

In addition, we have

$$\sum_{j:x_{\rho(j)}=1} \omega_j \alpha_j = \sum_{j:x_{\rho(j)}=1} \omega_j \mathbf{M}_j \mathbf{u} = \mathbf{1}\mathbf{u} = \alpha.$$

This means

$$\prod_{j:x_{\rho(j)}=1} \Big(e(C_0^{\mathbf{x}}, \boldsymbol{D}_j) \cdot e(C_1, D_{0,j})^{-1} \Big)^{\omega_j} = \prod_{j:x_{\rho(j)}=1} e(g_1, g_1)^{\omega_j \alpha_j s} = e(g_1, g_1)^{\alpha s}.$$

Correctness follows readily.

3.2 Proof of Security

We prove the following theorem:

Theorem 1. *Under Assumptions 1, 2 and 3 (described in Section 2.4), our KP-ABE scheme defined in Section 3.1 is semi-adaptively secure (in the sense of Definition 2). More precisely, for any adversary $\mathcal{A}$ that makes at most q key queries against the KP-ABE scheme, there exist probabilistic algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\text{KP-ABE}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_1}^{\text{AS1}}(\lambda) + n \cdot \mathsf{Adv}_{\mathcal{B}_2}^{\text{AS3}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\text{AS2}}(\lambda) + 1/p_1 + (n+1)/p_2,$$

and

$$\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3)\} \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n),$$

where n is the size of universe attribute set and $\mathrm{poly}(\lambda, n)$ is independent of $\mathsf{Time}(\mathcal{A})$.

[5] It is easy to see that $e(C_0^{\mathbf{x}}, \boldsymbol{D}_j)$ can in fact be computed using only a single pairing.

Overview. The proof follows via a series of games. To describe the games, we must first define semi-functional keys and ciphertexts. Fix random generators g_1, g_2, g_3, and let $\mathbf{x}^*$ denote the semi-adaptive challenge. We stress that unlike standard dual system encryption, we allow the semi-functional secret keys to depend on the semi-adaptive challenge $\mathbf{x}^*$ (this is okay because in the semi-adaptive security game, $\mathbf{x}^*$ is fixed before the adversary sees any secret keys). In the final transition (c.f. Lemma 3), we need to be able to simulate the secret keys given $g_1^\alpha X_2$ (as provided in Assumption 2) instead of g_1^α, so we define the semi-functional secret keys to have additional random G_{p_2}-components for the indices j corresponding to $x^*_{\rho(j)} = 0$ as captured by the term $\alpha'_j \mathbf{e}_{\rho(j)}$ below.

Semi-functional ciphertext.

$$\text{CT}_{\mathbf{x}^*} := \left(g_1^s \cdot \boxed{g_2^{s'}},\ g_1^{s\langle \mathbf{w}, \mathbf{x}^* \rangle} \cdot \boxed{g_2^{s'\langle \mathbf{w}, \mathbf{x}^* \rangle}},\ e(g_1, g_1)^{\alpha s} \cdot m \right),$$

where $s' \leftarrow_{\text{R}} \mathbb{Z}_N$.

Semi-functional secret key.

$$\text{SK}_{\mathbb{A}} := \begin{pmatrix} g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j : x^*_{\rho(j)} = 1 \\ g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{\boxed{\alpha'_j \mathbf{e}_{\rho(j)}} + r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j : x^*_{\rho(j)} = 0 \end{pmatrix},$$

where fresh $\alpha'_1, \ldots, \alpha'_\ell \leftarrow_{\text{R}} \mathbb{Z}_N$ are chosen for each secret key (specifically, we pick fresh $\alpha'_j \leftarrow_{\text{R}} Z_N$ for all j such that $x^*_{\rho(j)} = 0$).

Remark 1 (decryption capabilities). Fix $\mathbf{x}^*, \mathbb{A}$ such that $\mathbf{x}^*$ satisfies $\mathbb{A}$. Then,

- both semi-functional and normal secret key $\text{SK}_{\mathbb{A}}$ can decrypt a normal ciphertext $\text{CT}_{\mathbf{x}^*}$;
- a normal secret key $\text{SK}_{\mathbb{A}}$ can decrypt a semi-functional ciphertext $\text{CT}_{\mathbf{x}^*}$;
- a semi-functional secret key $\text{SK}_{\mathbb{A}}$ *can* decrypt a semi-functional ciphertext $\text{CT}_{\mathbf{x}^*}$; this is because the j'th subkey $(\boldsymbol{D}_j, D_{0,j})$ corresponding to $x^*_{\rho(j)} = 0$ is not used for decryption although it has an additional semi-functional component $g_2^{\alpha'_j}$. This is different from a standard dual system encryption argument, but is okay in our setting because $\mathbf{x}^*$ is fixed semi-adaptively *before* the adversary makes secret key queries.

Game Sequence. We consider the following sequence of games:

- Game_0: is the real security game (c.f. Section 2.3).
- Game_1: is the same as Game_0 except that the challenge ciphertext is semi-functional.

– $\mathsf{Game}_{2,k}$, $k = 1, 2, \ldots, n$: we incrementally transform each normal secret key to a semi-functional one, i.e. $\mathsf{Game}_{2,k}$ is the same as Game_1 except that, for each secret key

$$\text{SK}_{\mathbb{A}} := \left(\boldsymbol{D}_j, D_{0,j} \quad : j \in [\ell] \right),$$

the j'th subkey $(\boldsymbol{D}_j, D_{0,j})$ is semi-functional if $\rho(j) \leq k$, and normal if $\rho(j) > k$. More precisely, $\text{SK}_{\mathbb{A}}$ has the distribution

$$\begin{pmatrix} g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j : x^*_{\rho(j)} = 1 \\ g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j : (x^*_{\rho(j)} = 0) \wedge (\rho(j) > k) \\ g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{\alpha'_j \mathbf{e}_{\rho(j)} + r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j : (x^*_{\rho(j)} = 0) \wedge (\rho(j) \leq k) \end{pmatrix},$$

where fresh $\alpha'_1, \ldots, \alpha'_\ell \leftarrow_{\text{R}} \mathbb{Z}_N$ are chosen for each secret key. In other words, from $\mathsf{Game}_{2,k-1}$ to $\mathsf{Game}_{2,k}$, we modify the first component $\boldsymbol{D}_j$ of the j'th subkey for all j such that $\rho(j) = k$ (that is, corresponds to the variable x_k) as follows:

- if $x^*_k = 1$, leave it unchanged;
- if $x^*_k = 0$, change the semi-functional component from $g_2^{r'_j \mathbf{w}}$ to $g_2^{\alpha'_j \mathbf{e}_k + r'_j \mathbf{w}}$.

Note that in $\mathsf{Game}_{2,n}$, all keys are semi-functional.

– Game_3: is the same as $\mathsf{Game}_{2,n}$ except that the challenge ciphertext is a semi-functional encryption of a random message in G_T.

Fix an adversary $\mathcal{A}$. We write $\mathsf{Adv}_{\text{xx}}(\lambda)$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\text{xx}}$. It is easy to see that $\mathsf{Adv}_3(\lambda) = 0$, because the view of the adversary is Game_3 is independent of the challenge bit β. We complete the proof by establishing the following sequence of lemmas.

Lemma 1 (Normal to semi-functional ciphertext). *There exists an adversary $\mathcal{B}_1$ such that:*

$$|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \leq \mathsf{Adv}^{\text{AS1}}_{\mathcal{B}_1}(\lambda) + 1/p_1 + 1/p_2$$

and $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \text{poly}(\lambda, n)$ *where* $\text{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct an adversary $\mathcal{B}_1$ for Assumption 1 using $\mathcal{A}$. Recall that in Assumption 1, the adversary is given $D := (\mathbb{G}; g_1, U_1 U_2, g_3)$, along with T, where T is distributed as

$$g_1^s \quad \text{or} \quad g_1^s g_2^{s'}.$$

Here, $\mathcal{B}_1$ simulates Game_0 if $T := g_1^s$ and Game_1 if $T := g_1^s g_2^{s'}$. The quantity s, s' in the assumption will correspond the random exponents s, s' used in the ciphertext.

Specifically, $\mathcal{B}_1$ proceeds as follows:

Setup. $\mathcal{B}_1$ samples $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_n$, $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_N^n$ and outputs

$$\text{MPK} := (\ e(g_1, g_1^{\alpha}), g_1, g_1^{\mathbf{w}}\ ).$$

We note that

$$(\ \alpha, \mathbf{w}, g_1, U_1U_2, g_3; T\ )$$

is known to $\mathcal{B}_1$. The adversary $\mathcal{A}$ outputs a challenge $\mathbf{x}^* := (x_1^*, \ldots, x_n^*)$.

Challenge Ciphertext. Upon receiving two equal-length messages m_0 and m_1 from $\mathcal{A}$, $\mathcal{B}_1$ picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$ and outputs the semi-functional challenge ciphertext as:

$$\text{CT}_{\mathbf{x}^*} := \Big(\ T,\ T^{\langle \mathbf{w}, \mathbf{x}^* \rangle},\ e(T, g_1^{\alpha}) \cdot m_{\beta}\Big).$$

Now, suppose $T = g_1^s \cdot g_2^{s'}$, then,

$$T^{\langle \mathbf{w}, \mathbf{x}^* \rangle} := (g_1^s \cdot g_2^{s'})^{\langle \mathbf{w}, \mathbf{x}^* \rangle} = g_1^{s\langle \mathbf{w}, \mathbf{x}^* \rangle} g_2^{s'\langle \mathbf{w}, \mathbf{x}^* \rangle},$$
$$e(T, g_1^{\alpha}) := e(g_1^s \cdot g_2^{s'}, g_1^{\alpha}) = e(g_1, g_1)^{\alpha s}.$$

Now, if $s' = 0$ (i.e., $T = g_1^s$), this would indeed be a normal encryption. On the other hand, if $s' \leftarrow_{\mathrm{R}} \mathbb{Z}_N$ instead, this would indeed be a semi-functional encryption.

Key Queries. On input $\mathbb{A} := (\mathbf{M}, \rho)$, $\mathcal{B}_1$ needs to generate a normal key $\text{SK}_{\mathbb{A}}$, which has the distribution

$$\Big(\ \boldsymbol{D}_j := g_1^{\alpha_j \mathbf{e}_{\rho(j)}} \cdot (g_1^{r_j} \cdot g_2^{r'_j})^{\mathbf{w}} \cdot \boldsymbol{X}_j,\ D_{0,j} := (g_1^{r_j} \cdot g_2^{r'_j}) \cdot Z_j\ : j \in [\ell]\ \Big).$$

$\mathcal{B}_1$ picks $\tilde{r}_j \leftarrow_{\mathrm{R}} \mathbb{Z}_N$ for $j \in [\ell]$ and replaces $g_1^{r_j} \cdot g_2^{r'_j}$ with $(U_1U_2)^{\tilde{r}_j}$; then, it outputs

$$\text{SK}_{\mathbb{A}} := (\ g_1^{\alpha_j \mathbf{e}_{\rho(j)}} \cdot (U_1U_2)^{\tilde{r}_j \mathbf{w}} \cdot \boldsymbol{X}_j,\ (U_1U_2)^{\tilde{r}_j} \cdot Z_j \quad : j \in [\ell]\ ).$$

Observe that $(U_1U_2)^{\tilde{r}_j}$ is properly distributed as long as U_1U_2 is a generator of $G_{p_1p_2}$ (by the Chinese Remainder Theorem), which occurs with probability $1 - 1/p_1 - 1/p_2$.

We may therefore conclude that: $|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \leq \mathsf{Adv}^{\mathrm{AS1}}_{\mathcal{B}_1}(\lambda) + 1/p_1 + 1/p_2$. □

Lemma 2 (Normal to semi-functional keys). *For $k = 1, \ldots, n$, there exists an adversary $\mathcal{B}_2$ such that:*

$$|\mathsf{Adv}_{2,k-1}(\lambda) - \mathsf{Adv}_{2,k}(\lambda)| \leq \mathsf{Adv}^{\mathrm{AS3}}_{\mathcal{B}_2}(\lambda) + 1/p_2$$

and $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$. *(We note that* $\mathsf{Game}_{2,0}$ *is identical to* Game_1*.)*

Overview of proof. Fix k. We want to modify j'th subkey $(\boldsymbol{D}_j, D_{0,j})$ for all j such that $\rho(j) = k$ (that is, corresponds to the variable x_k) as follows:

- if $x^*_k = 1$, we leave it unchanged (in this case, $\mathsf{Game}_{2,k-1}$ and $\mathsf{Game}_{2,k}$ are identical);
- if $x^*_k = 0$, we change the semi-functional component in $\boldsymbol{D}_j$ from $g_2^{r'_j \mathbf{w}}$ to $g_2^{\alpha'_j \mathbf{e}_k + r'_j \mathbf{w}}$ using Assumption 3.

In the rest of the overview, we focus on the case $x^*_k = 0$. Roughly speaking, we rely on the fact that $w_k \pmod{p_2}$ is statistically hidden given MPK to obtain computational entropy as captured by $\{g_2^{\alpha'_j} : \rho(j) = k\}$. For simplicity, we first consider a single subkey $(\boldsymbol{D}_j, D_{0,j})$ for which $\rho(j) = k$. Recall that $(\boldsymbol{D}_j, D_{0,j})$ in $\mathsf{Game}_{2,k-1}$ and $\mathsf{Game}_{2,k}$ are of the form:

$$(g_1^{\alpha_j \mathbf{e}_k + r_j \mathbf{w}} \cdot \boxed{g_2^{r'_j \mathbf{w}}} \cdot \boldsymbol{X}_j,\ g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j) \quad \text{and}$$
$$(g_1^{\alpha_j \mathbf{e}_k + r_j \mathbf{w}} \cdot \boxed{g_2^{\alpha'_j \mathbf{e}_k + r'_j \mathbf{w}}} \cdot \boldsymbol{X}_j,\ g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j)$$

Roughly speaking, it suffices to show that:

$$(g_1^{\mathbf{w}},\ \boxed{g_2^{r'_j \mathbf{w}}} \cdot \boldsymbol{X}_j,\ g_2^{r'_j} \cdot Z_j) \quad \text{and} \quad (g_1^{\mathbf{w}},\ \boxed{g_2^{\alpha'_j \mathbf{e}_k + r'_j \mathbf{w}}} \cdot \boldsymbol{X}_j,\ g_2^{r'_j} \cdot Z_j)$$

are computationally indistinguishable, where $g_1^{\mathbf{w}}$ is provided in MPK. We may further simplify this to show that:

$$(g_1^{w_k},\ \boxed{g_2^{r'_j w_k}} \cdot X_j,\ g_2^{r'_j} \cdot Z_j) \quad \text{and} \quad (g_1^{w_k},\ \boxed{g_2^{\alpha'_j + r'_j w_k}} \cdot X_j,\ g_2^{r'_j} \cdot Z_j)$$

are computationally indistinguishable, where $X_j, Z_j \leftarrow_{\mathrm{R}} G_{p_3}$. This follows essentially from Assumption 3, which tells us that

$$(\boxed{g_2^{r'_j w_k}} \cdot X_j,\ g_2^{r'_j} \cdot Z_j,\ g_2^{w_k} \cdot Y_3) \quad \text{and} \quad (\boxed{g_2^{\alpha'_j + r'_j w_k}} \cdot X_j,\ g_2^{r'_j} \cdot Z_j,\ g_2^{w_k} \cdot Y_3)$$

are computationally indistinguishable, where $X_j, Z_j, Y_3 \leftarrow G_{p_3}$. Here, we rely crucially on the fact that $w_k \pmod{p_2}$ is completely random given $g_1^{w_k}$. To handle multiple subkeys $\{(\boldsymbol{D}_j, D_{0,j}) : j \in \rho^{-1}(k)\}$, we can proceed via a hybrid argument, but that would yield a security loss of $|\rho^{-1}(k)|$. To avoid this loss, we rely on the re-randomization trick from [28]. Finally, note that we cannot generate a semi-functional ciphertext for $\mathbf{x}^*$ such that $x^*_k = 1$ since we are only given $g_2^{w_k} Y_3$ and not $g_2^{w_k}$. (For the proof, it suffices to simulate a semi-functional ciphertext for which $x^*_k = 0$.)

Proof. We construct an adversary $\mathcal{B}_2$ (which gets as additional input $k \in [n]$) for Assumption 3 using $\mathcal{A}$. We note that the case $x^*_k = 1$ is straight-forward since $\mathsf{Game}_{2,k}$ is identical to $\mathsf{Game}_{2,k-1}$, which means

$$|\mathsf{Adv}_{2,k-1}(\lambda) - \mathsf{Adv}_{2,k}(\lambda)| = 0 \leq \mathsf{Adv}^{\mathrm{AS3}}_{\mathcal{B}_2}(\lambda).$$

This leaves us with k such that $x_k^* = 0$. Recall that in Assumption 3, the adversary is given $D := (\mathbb{G}; g_1, U_1U_2, g_2^x X_3, g_2^y Y_3, g_2 U_3, g_3)$, along with T, where T is distributed as

$$g_2^{xy} W_3 \quad \text{or} \quad g_2^{xy+z} W_3.$$

Here, we assume that $z \leftarrow_{\mathrm{R}} \mathbb{Z}_{p_2}^*$, which yields a $1/p_2$ negligible difference from Assumption 3 in the advantage; $\mathcal{B}_2$ simulates $\mathsf{Game}_{2,k-1}$ if $T = g_2^{xy} W_3$ and $\mathsf{Game}_{2,k}$ if $T = g_2^{xy+z} W_3$. Moreover, we use a "trick" from [28] to get a tight security reduction and avoid losing a factor of ℓ.

Specifically, $\mathcal{B}_2$ proceeds as follows:

Setup. $\mathcal{B}_2$ samples $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_N$, $\tilde{\mathbf{w}} \leftarrow_{\mathrm{R}} \mathbb{Z}_N^n$ and implicitly sets the parameter $\mathbf{w} := \tilde{\mathbf{w}} \bmod p_1 p_3$ (whereas $\mathbf{w} \bmod p_2$ is undetermined at this point). $\mathcal{B}_2$ outputs

$$\text{MPK} := (\ e(g_1, g_1^\alpha), g_1, g_1^{\tilde{\mathbf{w}}}\ ).$$

Observe that this is indeed the correct distribution since $g_1^{\mathbf{w}} = g_1^{\tilde{\mathbf{w}}}$. Moreover, we note that

$$(\ \alpha, \tilde{\mathbf{w}}, g_3; U_1U_2, g_2^x X_3, g_2^y Y_3, g_2 U_3; T\ )$$

is known to $\mathcal{B}_2$. Upon receiving a challenge $\mathbf{x}^* := (x_1^*, \ldots, x_n^*)$ for which $x_k^* = 0$, $\mathcal{B}_2$ implicitly sets the parameter $\mathbf{w} = \tilde{\mathbf{w}} + y \cdot \mathbf{e}_k \bmod p_2$.

Challenge Ciphertext. Upon receiving two equal-length messages m_0 and m_1 from $\mathcal{A}$, $\mathcal{B}_2$ picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$ and outputs the semi-functional challenge ciphertext as:

$$\Big(\ U_1U_2,\ (U_1U_2)^{\langle \tilde{\mathbf{w}}, \mathbf{x}^* \rangle},\ e(g_1^\alpha, U_1U_2) \cdot m_\beta\ \Big).$$

Observe that this is indeed the correct distribution since $\langle \tilde{\mathbf{w}}, \mathbf{x}^* \rangle = \langle \mathbf{w}, \mathbf{x}^* \rangle \bmod p_1 p_2$.

Key Queries. On input $\mathbb{A} := (\mathbf{M}, \rho)$, $\mathcal{B}_2$ needs to generate a secret key $\text{SK}_{\mathbb{A}}$ of the form:

$$\left(\begin{array}{ll}
g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{r_j' \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r_j'} \cdot Z_j \ : x_{\rho(j)}^* = 1 \\
g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{r_j' \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r_j'} \cdot Z_j \ : (x_{\rho(j)}^* = 0) \wedge (\rho(j) > k) \\
g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot g_2^{\alpha_j' \mathbf{e}_{\rho(j)} + r_j' \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r_j'} \cdot Z_j \ : (x_{\rho(j)}^* = 0) \wedge (\rho(j) < k) \\
g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot \boxed{g_2^{r_j' \mathbf{w}}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r_j'} \cdot Z_j \ : (x_{\rho(j)}^* = 0) \wedge (\rho(j) = k) \\
 & \qquad \wedge (T = g_2^{xy} W_3) \\
g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \mathbf{w}} \cdot \boxed{g_2^{\alpha_j' \mathbf{e}_{\rho(j)} + r_j' \mathbf{w}}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r_j'} \cdot Z_j \ : (x_{\rho(j)}^* = 0) \wedge (\rho(j) = k) \\
 & \qquad \wedge (T = g_2^{xy+z} W_3)
\end{array}\right)$$

where $\alpha_1', \ldots, \alpha_\ell' \leftarrow_{\mathrm{R}} \mathbb{Z}_N$. Note that we know α and can therefore compute $\alpha_j := \mathbf{M}_j \mathbf{u}$ as in the normal KeyGen. We proceed via a case analysis for j. The first three cases are straight-forward, observe that

$$g_1^{\tilde{\mathbf{w}}} = g_1^{\mathbf{w}} \quad \text{and} \quad g_2^{\mathbf{w}} = g_2^{\tilde{\mathbf{w}}} \cdot (g_2^y)^{\mathbf{e}_k}.$$

We simply use g_2U_3 and $g_2^yY_3$ in place of g_2 and g_2^y respectively and pick $r_j, r'_j, \alpha'_j \leftarrow_{\mathrm{R}} \mathbb{Z}_N$.
This leaves us with j such that $(x^*_{\rho(j)} = 0) \wedge (\rho(j) = k)$. Here, $\mathcal{B}_2$ picks $\delta_j, \delta'_j \leftarrow_{\mathrm{R}} \mathbb{Z}_N$ and implicitly sets

$$r'_j := x\delta_j + \delta'_j.$$

We can then rewrite the j'th normal subkey as:

$$\big(\, g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \tilde{\mathbf{w}}} \cdot \boxed{(g_2^{x\delta_j} \cdot g_2^{\delta'_j})^{\tilde{\mathbf{w}}} \cdot (g_2^{xy\delta_j} \cdot g_2^{y\delta'_j})^{\mathbf{e}_{\rho(j)}}} \cdot \boldsymbol{X}_j,\ g_1^{r_j} \cdot (g_2^{x\delta_j} \cdot g_2^{\delta'_j}) \cdot Z_j \,\big).$$

Here, we want to replace $g_2, g_2^x, g_2^y, g_2^{xy}$ with $g_2U_3, g_2^xX_3, g_2^yY_3, T$ respectively. First, $\mathcal{B}_2$ computes

$$R_j := (g_2^x X_3)^{\delta_j} \cdot (g_2 U_3)^{\delta'_j} = g_2^{r'_j} \cdot (X_3^{\delta_j} U_3^{\delta'_j}),$$

and outputs as the j'th subkey

$$\big(\, g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \tilde{\mathbf{w}}} \cdot \boxed{R_j^{\tilde{\mathbf{w}}} \cdot \big(T^{\delta_j} \cdot (g_2^y Y_3)^{\delta'_j}\big)^{\mathbf{e}_{\rho(j)}}} \cdot \boldsymbol{X}_j,\ g_1^{r_j} \cdot R_j \cdot Z_j \,\big).$$

Now, suppose $T = g_2^{xy+z} W_3$. Then,

$$R_j^{\tilde{\mathbf{w}}} \cdot \big(T^{\delta_j} \cdot (g_2^y Y_3)^{\delta'_j}\big)^{\mathbf{e}_{\rho(j)}} = g_2^{z\delta_j \mathbf{e}_{\rho(j)} + r'_j \mathbf{w}} \cdot \boldsymbol{X}'_j$$

for some $\boldsymbol{X}'_j \in G_{p_3}^n$. Now, if $z = 0$ (i.e., $T = g_2^{xy} W_3$), this would indeed be a normal subkey. On the other hand, if $z \leftarrow_{\mathrm{R}} \mathbb{Z}_{p_2}^*$, this would be a semi-functional subkey, with $\alpha'_j := z\delta_j$, and where (r'_j, δ_j) are pairwise-independent modulo p_2.
In summary, $\mathcal{B}_2$ outputs as $\mathrm{SK}_{\mathbb{A}}$:

$$\begin{pmatrix}
\tilde{\boldsymbol{D}}_j \cdot \boldsymbol{S}_j, & \tilde{D}_{0,j} \cdot (g_2 U_3)^{r'_j} : x^*_{\rho(j)} = 1 \\
\tilde{\boldsymbol{D}}_j \cdot \boldsymbol{S}_j, & \tilde{D}_{0,j} \cdot (g_2 U_3)^{r'_j} : (x^*_{\rho(j)} = 0) \wedge (\rho(j) > k) \\
\tilde{\boldsymbol{D}}_j \cdot (g_2 U_3)^{\alpha'_j \mathbf{e}_{\rho(j)}} \cdot \boldsymbol{S}_j, & \tilde{D}_{0,j} \cdot (g_2 U_3)^{r'_j} : (x^*_{\rho(j)} = 0) \wedge (\rho(j) < k) \\
\tilde{\boldsymbol{D}}_j \cdot \boxed{R_j^{\tilde{\mathbf{w}}} \cdot \big(T^{\delta_j} \cdot (g_2^y Y_3)^{\delta'_j}\big)^{\mathbf{e}_{\rho(j)}}}, & \tilde{D}_{0,j} \cdot R_j \quad : (x^*_{\rho(j)} = 0) \wedge (\rho(j) = k)
\end{pmatrix}$$

where

$$\tilde{\boldsymbol{D}}_j := g_1^{\alpha_j \mathbf{e}_{\rho(j)} + r_j \tilde{\mathbf{w}}} \cdot \boldsymbol{X}_j \in G_{p_1 p_3}^n, \qquad \tilde{D}_{0,j} := g_1^{r_j} \cdot Z_j \in G_{p_1 p_3},$$
$$\boldsymbol{S}_j := (g_2^y Y_3)^{r'_j \mathbf{e}_k} \cdot (g_2 U_3)^{r'_j \tilde{\mathbf{w}}} \in G_{p_2 p_3}^n, \quad R_j := (g_2^x X_3)^{\delta_j} \cdot (g_2 U_3)^{\delta'_j} \in G_{p_2 p_3}.$$

We may therefore conclude that: $|\mathsf{Adv}_{2,k-1}(\lambda) - \mathsf{Adv}_{2,k}(\lambda)| \le \mathsf{Adv}^{\mathrm{AS3}}_{\mathcal{B}_2}(\lambda) + 1/p_2$. □

Lemma 3 (Final transition). *There exists an adversary $\mathcal{B}_3$ such that:*

$$|\mathsf{Adv}_{2,n}(\lambda) - \mathsf{Adv}_3(\lambda)| \le \mathsf{Adv}^{\mathrm{AS2}}_{\mathcal{B}_3}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Overview of proof. Following the final transitions in [25, 27], we use Assumption 2, in which we are given $(g_1, g_1^{\alpha} X_2, g_1^{s} Y_2, g_2, g_3, T)$ where T is either $e(g_1, g_1)^{\alpha s}$ or drawn uniformly from G_T to blind the challenge message m_β. The main challenge in our setting lies in simulating a semi-functional key $\text{SK}_{\mathbb{A}}$ given $g_1^{\alpha} X_2$ and not α itself. Recall that a semi-functional key $\text{SK}_{\mathbb{A}}$ has the same distribution

$$\begin{pmatrix} \boxed{g_1^{\alpha_j \mathbf{e}_{\rho(j)}}} \cdot g_1^{r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j & : x^*_{\rho(j)} = 1 \\ \boxed{g_1^{\alpha_j \mathbf{e}_{\rho(j)}}} \cdot \boxed{g_2^{\alpha'_j \mathbf{e}_{\rho(j)}}} \cdot g_1^{r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j & : x^*_{\rho(j)} = 0 \end{pmatrix}$$

in both $\mathsf{Game}_{2,n}$ and Game_3. Specifically, we need to simulate (given $g_1, g_2, g_1^{\alpha} X_2$)

$$\begin{pmatrix} \boxed{g_1^{\alpha_j}} & : x^*_{\rho(j)} = 1 \\ \boxed{g_1^{\alpha_j}} \cdot \boxed{g_2^{\alpha'_j}} & : x^*_{\rho(j)} = 0 \end{pmatrix}$$

where $\alpha_1, \ldots, \alpha_\ell$ are LSSS shares of α according to $\mathbb{A} = (\mathbf{M}, \rho)$ and $\alpha'_1, \ldots, \alpha'_\ell$ are independently random values. Roughly speaking, we proceed as follows:

- simulate the terms $(g_1^{\alpha_j} : x^*_{\rho(j)} = 1)$ by raising g_1 to the power of random LSSS shares of 0 (as determined by $\mathbf{M}\tilde{\mathbf{u}}_0$ below);
- simulate the terms $(g_1^{\alpha_j} \cdot g_2^{\alpha'_j} : x^*_{\rho(j)} = 0)$ by doing a LSSS share of $g_1^{\alpha} X_2$ "in the exponent" (as determined by $\alpha \mathbf{M}\tilde{\mathbf{u}}_1$ below), multiplying by the shares of 0 from the previous step, then re-randomizing the G_{p_2}-components.

We exploit the fact that $\mathbf{x}^*$ does not satisfy $\mathbb{A}$ to argue that we can choose $\tilde{\mathbf{u}}_1$ so that $\mathbf{M}_{\mathbf{x}^*} \tilde{\mathbf{u}}_1 = \mathbf{0}$.

Proof. We construct an adversary $\mathcal{B}_3$ for Assumption 2 using $\mathcal{A}$. Recall that in Assumption 2, the adversary is given $D := (\mathbb{G}; g_1, g_1^{\alpha} X_2, g_1^{s} Y_2, g_2, g_3)$, along with T, where T equals $e(g_1, g_1)^{\alpha s}$ or is drawn uniformly from G_T. Here, $\mathcal{B}_3$ simulates $\mathsf{Game}_{2,n}$ if $T := e(g_1, g_1)^{\alpha s}$ and Game_3 if $T \leftarrow_{\text{R}} G_T$. The quantity α in the assumption will correspond exactly to α in MSK, and the quantity s in the assumption will correspond the random exponents s used in the (semi-functional) ciphertext.

Specifically, $\mathcal{B}_3$ proceeds as follows:

Setup. $\mathcal{B}_3$ samples $\mathbf{w} \leftarrow_{\text{R}} \mathbb{Z}_N^n$ and output the public parameters

$$\text{MPK} := (\ e(g_1, g_1^{\alpha} X_2), g_1, g_1^{\mathbf{w}}\ ).$$

We note that

$$(\ \mathbf{w}, g_2, g_3; g_1^{\alpha} X_2, g_1^{s} Y_2; T\ )$$

is known to $\mathcal{B}_3$. The adversary $\mathcal{A}$ outputs a challenge $\mathbf{x}^* := (x_1^*, \ldots, x_n^*)$.

Challenge Ciphertext. Upon receiving two equal-length messages m_0 and m_1 from $\mathcal{A}$, $\mathcal{B}_3$ picks $\beta \leftarrow_{\text{R}} \{0, 1\}$ and outputs the semi-functional challenge ciphertext as:

$$\text{CT}_{\mathbf{x}^*} := \Big(\ g_1^{s} Y_2,\ (g_1^{s} Y_2)^{\langle \mathbf{w}, \mathbf{x}^* \rangle},\ T \cdot m_\beta\ \Big).$$

Now, if T is distributed as distributed as $e(g_1, g_1)^{\alpha s}$, this would indeed be a properly distributed semi-functional encryption of m_β. On the other hand, if $T \leftarrow_{\mathrm{R}} G_T$, instead, then the challenge ciphertext is a properly distributed semi-functional encryption of a random message in G_T.

Key Queries. On input $\mathbb{A} := (\mathbf{M}, \rho)$, $\mathcal{B}_3$ needs to generate a semi-functional key $\mathrm{SK}_{\mathbb{A}}$, which has the distribution

$$\begin{pmatrix} \boxed{g_1^{\alpha_j \mathbf{e}_{\rho(j)}}} \cdot g_1^{r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j & : x^*_{\rho(j)} = 1 \\ \boxed{g_1^{\alpha_j \mathbf{e}_{\rho(j)}}} \cdot \boxed{g_2^{\alpha'_j \mathbf{e}_{\rho(j)}}} \cdot g_1^{r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j, & g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j & : x^*_{\rho(j)} = 0 \end{pmatrix},$$

where $\alpha'_1, \ldots, \alpha'_\ell \leftarrow_{\mathrm{R}} \mathbb{Z}_N$. The main challenge lies in simulating the terms $g_1^{\alpha_j}$ since $\mathcal{B}_3$ is only given $g_1^\alpha X_2$ and not α itself. By definition of the KP-ABE security game, $\mathbf{x}^*$ does not satisfy $\mathbb{A}$, so $\mathbf{1} \notin \mathsf{span}\langle \mathbf{M}_{\mathbf{x}^*} \rangle$. (Refer to Definition 1 for the notation.) Therefore, we can efficiently compute $\tilde{\mathbf{u}}_1 \in \mathbb{Z}_N^{\ell'}$ such that

$$\mathbf{M}_{\mathbf{x}^*} \tilde{\mathbf{u}}_1 = \mathbf{0} \quad \text{and} \quad \mathbf{1} \tilde{\mathbf{u}}_1 = 1.$$

$\mathcal{B}_3$ samples $\tilde{\mathbf{u}}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_N^{\ell'}$ such that $\mathbf{1} \tilde{\mathbf{u}}_0 = 0$, and implicitly sets

$$\mathbf{u} := \alpha \cdot \tilde{\mathbf{u}}_1 + \tilde{\mathbf{u}}_0.$$

Observe that $\mathbf{u}$ has indeed the correct distribution. Recall that we set $\alpha_j := \mathbf{M}_j \mathbf{u}$, which yields

$$\alpha_j = \begin{cases} \mathbf{M}_j \tilde{\mathbf{u}}_0 & \text{if } x^*_{\rho(j)} = 1 \\ \alpha \cdot \mathbf{M}_j \tilde{\mathbf{u}}_1 + \mathbf{M}_j \tilde{\mathbf{u}}_0 & \text{if } x^*_{\rho(j)} = 0 \end{cases}$$

where both $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{u}}_0$ are known to $\mathcal{B}_3$. The case j such that $x^*_{\rho(j)} = 1$ is straight-forward; $\mathcal{B}_3$ simply picks $r_j, r'_j \leftarrow_{\mathrm{R}} \mathbb{Z}_N$. For the case j such that $x^*_{\rho(j)} = 0$, we can then rewrite $g_1^{\alpha_j} \cdot g_2^{\alpha'_j}$ as a function of $\tilde{\mathbf{u}}_0, \tilde{\mathbf{u}}_1$, and $g_1^\alpha X_2$:

$$g_1^{\alpha_j} \cdot g_2^{\alpha'_j} = g_1^{\alpha \cdot \mathbf{M}_j \tilde{\mathbf{u}}_1 + \mathbf{M}_j \tilde{\mathbf{u}}_0} \cdot g_2^{\alpha'_j} = (g_1^\alpha X_2)^{\mathbf{M}_j \tilde{\mathbf{u}}_1} \cdot g_1^{\mathbf{M}_j \tilde{\mathbf{u}}_0} \cdot g_2^{\tilde{\alpha}'_j},$$

where $\mathcal{B}_3$ picks $\tilde{\alpha}'_j \leftarrow_{\mathrm{R}} \mathbb{Z}_N$ and implicitly sets $g_2^{\alpha'_j} := X_2^{\mathbf{M}_j \tilde{\mathbf{u}}_1} \cdot g_2^{\tilde{\alpha}'_j}$. $\mathcal{B}_3$ then outputs

$$\mathrm{SK}_{\mathbb{A}} := \begin{pmatrix} \boxed{g_1^{\mathbf{M}_j \tilde{\mathbf{u}}_0 \mathbf{e}_{\rho(j)}}} \cdot \tilde{\boldsymbol{D}}_j, & D_{0,j} & : x^*_{\rho(j)} = 1 \\ \boxed{\left((g_1^\alpha X_2)^{\mathbf{M}_j \tilde{\mathbf{u}}_1} \cdot g_1^{\mathbf{M}_j \tilde{\mathbf{u}}_0} \cdot g_2^{\tilde{\alpha}'_j}\right)^{\mathbf{e}_{\rho(j)}}} \cdot \tilde{\boldsymbol{D}}_j, & D_{0,j} & : x^*_{\rho(j)} = 0 \end{pmatrix},$$

where $\tilde{\boldsymbol{D}}_j := g_1^{r_j \mathbf{w}} \cdot g_2^{r'_j \mathbf{w}} \cdot \boldsymbol{X}_j$ and $\tilde{D}_{0,j} := g_1^{r_j} \cdot g_2^{r'_j} \cdot Z_j$.

We may therefore conclude that: $|\mathsf{Adv}_{2,n}(\lambda) - \mathsf{Adv}_3(\lambda)| \leq \mathsf{Adv}^{\mathrm{AS2}}_{\mathcal{B}_3}(\lambda)$. □

Acknowledgments. We thank Allison Lewko and the reviewers for helpful discussions and feedback.

References

[1] Attrapadung, N.: Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577. Springer, Heidelberg (2014)

[2] Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011)

[3] Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. Ph.D., Technion - Israel Institute of Technology (1996)

[4] Benabbas, S., Gennaro, R., Vahlis, Y.: Verifiable delegation of computation over large datasets. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 111–131. Springer, Heidelberg (2011)

[5] Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM J. Comput. 32(3), 586–615 (2003)

[6] Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)

[7] Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)

[8] Chen, J., Wee, H.: Fully (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (2013)

[9] Chen, J., Wee, H.: Fully (almost) tightly secure IBE from standard assumptions. IACR Cryptology ePrint Archive, Report 2013/803, Preliminary version in [8] (2013)

[10] Chen, J., Wee, H.: Dual system groups and its applications — compact HIBE and more. IACR Cryptology ePrint Archive, Report 2014/265, Preliminary version in [8] (2014)

[11] Chen, J., Wee, H.: Semi-adaptive attribute-based encryption and improved delegation for boolean formula. IACR Cryptology ePrint Archive, Report 2014/465 (2014)

[12] Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter IBE and signatures via asymmetric pairings. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 122–140. Springer, Heidelberg (2013)

[13] Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: ACM Conference on Computer and Communications Security, pp. 456–465 (2007)

[14] Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)

[15] Fiore, D., Gennaro, R.: Publicly verifiable delegation of large polynomials and matrix computations, with applications. In: ACM Conference on Computer and Communications Security, pp. 501–512 (2012)

[16] Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 44–61. Springer, Heidelberg (2010)
[17] Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 465–482. Springer, Heidelberg (2010)
[18] Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645. Springer, Heidelberg (2013)
[19] Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: Delegating computation: Interactive proofs for muggles. In: STOC, pp. 113–122 (2008)
[20] Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
[21] Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
[22] Karchmer, M., Wigderson, A.: On span programs. In: Structure in Complexity Theory Conference, pp. 102–111 (1993)
[23] Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
[24] Lewko, A.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 318–335. Springer, Heidelberg (2012)
[25] Lewko, A., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
[26] Lewko, A., Waters, B.: New proof methods for attribute-based encryption: Achieving full security through selective techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012)
[27] Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
[28] Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. J. ACM 51(2), 231–262 (2004)
[29] Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 57–74. Springer, Heidelberg (2008)
[30] Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer, Heidelberg (2009)
[31] Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
[32] Parno, B., Raykova, M., Vaikuntanathan, V.: How to delegate and verify in public: Verifiable computation from attribute-based encryption. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 422–439. Springer, Heidelberg (2012)

[33] Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
[34] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) Advances in Cryptology - CRYPT0 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
[35] Takashima, K.: Expressive attribute-based encryption with constant-size ciphertexts from the decisional linear assumption. In: SCN Also, Cryptology ePrint Archive, Report 2014/207 (to appear 2014)
[36] Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
[37] Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
[38] Wee, H.: Dual system encryption via predicate encodings. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 616–637. Springer, Heidelberg (2014)