Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts

Nuttapong Attrapadung[1], Benoît Libert[2*], and Elie de Panafieu[3]

[1] Research Center for Information Security, AIST (Japan)
[2] Université catholique de Louvain, ICTEAM – Crypto Group (Belgium)
[3] Ecole Normale Supérieure, Cachan (France)

Abstract. Attribute-based encryption (ABE), as introduced by Sahai and Waters, allows for fine-grained access control on encrypted data. In its key-policy flavor, the primitive enables senders to encrypt messages under a set of attributes and private keys are associated with access structures that specify which ciphertexts the key holder will be allowed to decrypt. In most ABE systems, the ciphertext size grows linearly with the number of ciphertext attributes and the only known exceptions only support restricted forms of threshold access policies.
This paper proposes the first key-policy attribute-based encryption (KP-ABE) schemes allowing for *non-monotonic* access structures (*i.e.*, that may contain negated attributes) and with constant ciphertext size. Towards achieving this goal, we first show that a certain class of identity-based broadcast encryption schemes generically yields monotonic KP-ABE systems in the selective set model. We then describe a new efficient identity-based revocation mechanism that, when combined with a particular instantiation of our general monotonic construction, gives rise to the first truly expressive KP-ABE realization with constant-size ciphertexts. The downside of these new constructions is that private keys have quadratic size in the number of attributes. On the other hand, they reduce the number of pairing evaluations to a constant, which appears to be a unique feature among expressive KP-ABE schemes.

Keywords. Attribute-based encryption, expressivity, efficiency.

1 Introduction

It frequently happens that sensitive data must be archived by storage servers in such a way that only specific parties are allowed to read the content. In these situations, enforcing the access control using ordinary public key encryption schemes is not very convenient as such primitives severely decrease the flexibility of users to share their data.

To address these concerns, Sahai and Waters [29] introduced attribute-based encryption (ABE), which refines identity-based encryption [30, 8] by associating ciphertexts and private keys with sets of descriptive attributes. Decryption is then possible when there is a sufficient overlap between the two sets. These results were extended by Goyal, Pandey, Sahai and Waters [20] into richer kinds of attribute-based encryption, where decryption is permitted when the attribute set satisfies a more complex boolean formula specified by an access structure. This paper describes truly expressive ABE systems featuring compact ciphertexts, regardless of the number of underlying attributes.

Related Work. Attribute-based encryption comes in two flavors. In key-policy ABE schemes (KP-ABE), attribute sets are used to annotate ciphertexts and private keys are associated with access structures that specify which ciphertexts the user will be entitled to decrypt. Ciphertext-policy ABE (CP-ABE) proceeds in the dual way, by assigning attribute sets to private keys and letting senders specify an access policy that receivers' attribute sets should comply with.

The ciphertext-policy scenario was first studied in [5, 18]. The construction of [18] only handles

[*] This author acknowledges the Belgian National Fund for Scientific Research (F.R.S.-F.N.R.S.) for his "chargé de recherches" fellowship and the BCRYPT Interuniversity Attraction Pole.

AND gates while the first expressive construction [5] was only analyzed in the generic group model. Goyal, Jain, Pandey and Sahai [21] gave a construction in the standard model but its large parameters and key sizes make it impractical for reasonably expressive policies. Efficient and expressive realizations in the standard model were subsequently put forth by Waters [32] and one of them was recently extended by Lewko *et al.* [25], and subsequently by Okamoto and Takashima [31], into schemes providing adaptive security whereas all prior works on ABE were limited to deal with selective adversaries [13, 14, 6] – who have to make up their mind about their target before having seen public parameters – in their security analysis.

In both CP-ABE and KP-ABE schemes, expressivity requires to go beyond what monotonic access structures can express. Ostrovsky, Sahai and Waters [28] considered access structures that may contain negative attributes without blowing up the size of shares or ciphertexts. Their initial construction was recently improved by Lewko, Sahai and Waters [24] who used techniques from revocation systems (which can be seen as negative analogues of identity-based broadcast encryption) to design the most efficient non-monotonic KP-ABE to date.

Our Contributions. So far, the research community has mostly focused on the design of expressive schemes – where access structures can implement as complex boolean formulas as possible – without trying to minimize the size of ciphertexts. Indeed, most schemes [20, 28, 32, 25, 24] feature linear-size ciphertexts in the maximal number of attributes that ciphertexts can be annotated with. In the ciphertext-policy setting, Emura *et al.* suggested a scheme with short ciphertexts [19] but policies are restricted to a single AND gate. More recently, Herranz *et al.* [22] described a scheme with threshold access policies and constant-size[4] ciphertexts. Yet, their scheme is still not as expressive as one could hope for. In particular, it seems difficult to extend it to support general linear-secret-sharing-realizable (or LSSS-realizable for short) access structures.

In the context of key-policy attribute-based encryption, this paper aims at devising schemes with constant-size ciphertexts[5] (regardless of the number of ciphertext attributes) allowing for as expressive policies as possible. To this end, we first show that a certain class of identity-based broadcast encryption (IBBE) schemes readily yields KP-ABE schemes with monotonic (though LSSS-realizable) access structures via a generic transformation. The latter preserves the ciphertext size and guarantees the resulting scheme to be selectively secure (as defined in [13, 6]) as long as the underlying IBBE system is itself selectively secure. At the expense of quadratic-size private keys (which comprise $O(t \cdot n)$ elements, where n is the maximal number of ciphertext attributes and t is the maximal number of leaf attributes in access trees), this transformation directly provides us with monotonic KP-ABE schemes with $O(1)$-size ciphertexts.

In a second step, we use a particular output of the aforementioned transformation to design a scheme supporting non-monotonic access structures without sacrificing the efficiency. In the resulting construction, the ciphertext overhead reduces to three group elements, no matter how many attributes ciphertexts are associated with. As in the monotonic case, private keys are inflated by a factor of n in comparison with [28, 24]. Nevertheless, these new schemes remain attractive for applications where bandwidth is the primary concern. In mobile Internet connections for instance, users are charged depending on the amount of transmitted messages; while in contrast, the storage is becoming much cheaper nowadays even for a large amount, as evidently in many smart phones.

As an intermediate step towards the new non-monotonic ABE, we design a new identity-based

[4] By "constant", we mean that the size only depends on the security parameter λ (the number of transmitted bits is typically $O(\lambda)$) and not on the number of ciphertext attributes.

[5] As in the literature on broadcast encryption (see, e.g., [9]) where the list of receivers is not included in the ciphertext, we do not count the description of ciphertext attributes as being part of the ciphertext. Indeed, many ciphertexts may have to be encrypted under the same attribute set.

revocation (IBR) mechanism (as defined by Lewko, Sahai and Waters [24]) with $O(1)$-size ciphertexts and a similar structure to that of the monotonic KP-ABE schemes provided by our general construction. This was necessary since prior IBR systems with short ciphertexts [4] were not directly amenable to fulfill these requirements. We believe this new IBR realization to be of independent interest since it performs noticeably better than previous schemes featuring short ciphertexts [4] and still relies a natural (though "q-type") intractability assumption.

The security of our schemes is proved against selective adversaries (that are not allowed to choose their target attribute set adaptively) under a non-interactive assumption. We leave it as an open problem to obtain KP-ABE schemes with compact ciphertexts that can be proven secure against adaptive adversaries (as in the work of Lewko *et al.* [25]).

Other Related Work. The aforementioned realizations all assume ABE schemes with a single authority and we focus on this context as well. Extensions to the multi-authority scenario were investigated in [15, 16] for a conjunctive setting and in [3] for a disjunctive setting. Besides the two usual flavors of ABE, another recently considered kind of ABE schemes [2], called dual-policy ABE, mixes features from both KP-ABE and CP-ABE systems.

Organization. In the following, we first review various primitives in section 2. Section 3 describes our general construction of monotonic KP-ABE. The new revocation scheme is depicted in section 4. Section 5 finally presents the non-monotonic ABE realization with compact ciphertexts.

2 Background and Definitions

Notation. We will treat a vector as a column vector, unless stated otherwise. Namely, for any vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{Z}_p^n$, $g^{\boldsymbol{\alpha}}$ stands for the vector of group elements $(g^{\alpha_1}, \ldots, g^{\alpha_n})^\top \in \mathbb{G}^n$. For $\boldsymbol{a}, \boldsymbol{z} \in \mathbb{Z}_p^n$, we denote their inner product as $\langle \boldsymbol{a}, \boldsymbol{z} \rangle = \boldsymbol{a}^\top \boldsymbol{z} = \sum_{i=1}^n a_i z_i$. Given $g^{\boldsymbol{a}}$ and $\boldsymbol{z}$, $(g^{\boldsymbol{a}})^{\boldsymbol{z}} := g^{\langle \boldsymbol{a}, \boldsymbol{z} \rangle}$ is computable without knowing $\boldsymbol{a}$. We denote by I_n the identity matrix of size n. For a set U, we define $2^U = \{S \mid S \subseteq U\}$ and $\binom{U}{<k} = \{S \mid S \subseteq U, |S| < k\}$ for $k \leq |U|$.

2.1 Syntax and Security Definition for Functional Encryption

We capture notions of KP-ABE, IBBE, IBR by providing a unified definition and security notion for functional encryption[6] here and then instantiating to these primitives in the next subsections.

Syntax. Let $R : \varSigma_k \times \varSigma_e \to \{0, 1\}$ be a boolean function where $\varSigma_k$ and $\varSigma_e$ denote "key index" and "ciphertext index" spaces. A functional encryption (FE) scheme for the relation R consists of algorithms: Setup, KeyGen, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$.

$\mathsf{Setup}(\lambda, \mathrm{des}) \to (\mathsf{mpk}, \mathsf{msk})$: The setup algorithm takes as input a security parameter λ and a scheme description des and outputs a master public key mpk and a master secret key msk.

$\mathsf{KeyGen}(\mathsf{msk}, X) \to \mathsf{sk}_X$: The key generation algorithm takes in the master secret key msk and a key index $X \in \varSigma_k$. It outputs a private key sk_X.

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, Y) \to C$: This algorithm takes as input a public key mpk, the message M, and a ciphertext index $Y \in \varSigma_e$. It outputs a ciphertext C.

$\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_X, X, C, Y) \to \mathsf{M}$ or $\perp$: The decryption algorithm takes in the public parameters mpk, a private key sk_X for the key index X and a ciphertext C for the ciphertext index Y. It outputs the message M or a symbol $\perp$ indicating that the ciphertext is not in a valid form.

[6] The term "functional encryption" was defined in slightly different manners in [25, 4, 31] before recently fully formalized in [11]. Our definition of FE here and throughout the paper refers to the class of predicate encryption with public index of [11].

Correctness mandates that, for all λ, all $(\mathsf{mpk}, \mathsf{msk})$ produced by $\mathsf{Setup}(\lambda, des)$, all $X \in \Sigma_k$, all keys sk_X returned by $\mathsf{KeyGen}(\mathsf{msk}, X)$ and all $Y \in \Sigma_e$,

– If $R(X, Y) = 1$, then $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, Y)), \mathsf{sk}_X) = \mathsf{M}$.
– If $R(X, Y) = 0$, then $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, Y)), \mathsf{sk}_X) = \perp$.

Security Notion. We now give the standard security definition for FE schemes.

Definition 1. *A FE scheme for relation R is fully secure if no probabilistic polynomial time (PPT) adversary $\mathcal{A}$ has non-negligible advantage in this game:*

Setup. *The challenger runs* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\lambda, \mathrm{des})$ *and gives* mpk *to* $\mathcal{A}$.

Phase 1. *On polynomially-many occasions, $\mathcal{A}$ chooses a key index X and gets* $\mathsf{sk}_X = \mathsf{Keygen}(\mathsf{msk}, X)$. *Such queries can be adaptive in that each one may depend on the information gathered so far.*

Challenge. *$\mathcal{A}$ chooses messages* $\mathsf{M}_0, \mathsf{M}_1$ *and a ciphertext index $Y^\star$ such that $R(X, Y^\star) = 0$ for all key indexes X that have been queried at step 2. Then, the challenger flips a fair binary coin $d \in \{0, 1\}$, generates a ciphertext* $C^\star = \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}_d, Y^\star)$, *and hands it to the adversary.*

Phase 2. *$\mathcal{A}$ is allowed to make more key generation queries for any key index X such that $R(X, Y^\star) = 0$.*

Guess. *$\mathcal{A}$ outputs a bit $d' \in \{0, 1\}$ and wins if $d' = d$.*

The advantage of the adversary $\mathcal{A}$ is measured by $\mathbf{Adv}(\lambda) := |\Pr[d' = d] - \frac{1}{2}|$.

A weaker notion called selective security [13, 6] can be defined as in the above game with the exception that the adversary $\mathcal{A}$ has to choose the challenge ciphertext index $Y^\star$ before the setup phase but private key queries $X_1, \ldots, X_q$ can still be adaptive. A dual notion called co-selective security [4], in contrast, requires $\mathcal{A}$ to declare q key queries for key indexes $X_1, \ldots, X_q$ before the setup phase, but $\mathcal{A}$ can adaptively choose the target challenge ciphertext index $Y^\star$.

2.2 Key-Policy Attribute-Based Encryption

Before describing the definition of KP-ABE, we first recall the definitions of access structures and linear secret sharing schemes, as defined in [20].

Definition 2 (Access Structures). *Consider a set of parties $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$. A collection $\mathbb{A} \subseteq 2^{\mathcal{P}}$ is said to be monotone if, for all B, C, if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure (resp., monotonic access structure) is a collection (resp., monotone collection) $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.*

Definition 3 (Linear Secret Sharing Scheme). *Let $\mathcal{P}$ be a set of parties. Let L be a $\ell \times k$ matrix. Let $\pi : \{1, \ldots, \ell\} \to \mathcal{P}$ be a function that maps a row to a party for labeling. A secret sharing scheme Π for access structure $\mathbb{A}$ over a set of parties $\mathcal{P}$ is a linear secret-sharing scheme (LSSS) in $\mathbb{Z}_p$ and is represented by (L, π) if it consists of two efficient algorithms:*

$\mathsf{Share}_{(L,\pi)}$**:** *takes as input $s \in \mathbb{Z}_p$ which is to be shared. It randomly chooses $\beta_2, \ldots, \beta_k \xleftarrow{R} \mathbb{Z}_p$ and let $\boldsymbol{\beta} = (s, \beta_2, \ldots, \beta_k)^\top$. It outputs $L \cdot \boldsymbol{\beta}$ as the vector of ℓ shares. The share $\lambda_i := \langle \boldsymbol{L_i}, \boldsymbol{\beta} \rangle$ belongs to party $\pi(i)$, where $\boldsymbol{L_i}^\top$ is the i^{th} row of L.*

$\mathsf{Recon}_{(L,\pi)}$**:** *takes as input an access set $S \in \mathbb{A}$. Let $I = \{i |\ \pi(i) \in S\}$. It outputs a set of constants $\{(i, \mu_i)\}_{i \in I}$ such that $\sum_{i \in I} \mu_i \cdot \lambda_i = s$.*

In a key-policy attribute-based encryption scheme, ciphertexts are associated with a set of attributes ω and private keys correspond to access structures $\mathbb{A}$. Decryption is possible when the attribute set ω is authorized in the access structure $\mathbb{A}$ (*i.e.*, $\omega \in \mathbb{A}$). We formally define it as an instance of FE as follows.

Definition 4 (KP-ABE). *Let U be an attribute space. Let $n \in \mathbb{N}$ be a bound on the number of attributes per ciphertext. A key-policy attribute-based encryption (KP-ABE) for a collection $\mathcal{AS}$ of access structures over U is a functional encryption for $R^{\mathsf{KP}} : \mathcal{AS} \times \binom{U}{<n} \to \{0,1\}$ defined by $R^{\mathsf{KP}}(\mathbb{A}, \omega) = 1$ iff $\omega \in \mathbb{A}$ (for $\omega \subseteq U$ such that $|\omega| < n$, and $\mathbb{A} \in \mathcal{AS}$). Furthermore, the description* des *consists of the attribute universe U, $\Sigma_k^{\mathsf{KP}} = \mathcal{AS}$, and $\Sigma_e^{\mathsf{KP}} = \binom{U}{<n}$.*

Definition 4 conforms with the original definition of KP-ABE, as in [20, 28, 24, 25, 11]. There is another variant of KP-ABE recently used in [31], that we call KP-ABE with labeling. We reformalize it in appendix A, for the purpose of comparison in Table 2. We remark that normal KP-ABE implies KP-ABE with labeling.

We note that chosen-ciphertext secure versions of our proposed KP-ABE schemes in this paper can be obtained from recent generic results of [33].

2.3 Identity-Based Broadcast Encryption and Revocation Scheme

An ID-based broadcast encryption, as formalized in [1], allows a sender to encrypt a message to a set of identities, say $S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_q\}$, where $q < n$ for some a-priori fixed bound $n \in \mathbb{N}$, so that a user who possesses a key for $\mathsf{ID} \in S$ can decrypt. In contrast, an ID-based revocation scheme [24] allows a sender to specify a revoked set S so that only a user with $\mathsf{ID} \notin S$ can decrypt.

Definition 5. *Let $\mathcal{I}$ be an identity space. An ID-based broadcast encryption scheme (IBBE) with the maximal bound n for the number of receivers per ciphertext is a functional encryption for $R^{\mathsf{IBBE}} : \mathcal{I} \times \binom{\mathcal{I}}{<n} \to \{0,1\}$ defined by $R^{\mathsf{IBBE}}(\mathsf{ID}, S) = 1$ iff $\mathsf{ID} \in S$.*

Definition 6. *Let $\mathcal{I}$ be an identity space. An ID-based revocation (IBR) with the maximal bound n for the number of revoked users per ciphertext is a functional encryption for $R^{\mathsf{IBR}} : \mathcal{I} \times \binom{\mathcal{I}}{<n} \to \{0,1\}$ defined by $R^{\mathsf{IBR}}(\mathsf{ID}, S) = 1$ iff $\mathsf{ID} \notin S$.*

Remark 1. Although selective and co-selective security are incomparable in general, we remark that, in IBR schemes, co-selective security implies selective security. To see why, we first recall that selective security for IBR requires the adversary $\mathcal{A}$ to declare the target revoked set $S^\star$ before seeing the public key mpk. Here, phase 1 can be simplified by letting the challenger hand over all the private keys for identities in $S^\star$ at once (along with mpk). On the other hand, co-selective IBR security requires $\mathcal{A}$ to declare the set $\tilde{S}$ of identities that will be queried for private key generation before seeing mpk whereas the target revocation set $S^\star$ does not have to be fully determined before the challenge phase. At the same time as mpk, the challenger then reveals all keys for identities in $\tilde{S}$ at once. Later, the adversary can choose any $S^\star \subseteq \tilde{S}$ in the challenge phase. Selective security corresponds to the special case where $S^\star = \tilde{S}$.

2.4 Complexity Assumptions

We use groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p with an efficiently computable mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ s.t. $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$, $a, b \in \mathbb{Z}$ and $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$. We rely on the DBDHE assumption introduced in [9]. This assumption is shown to hold in the generic group model [7]. In addition, it is non-interactive and falsifiable [26].

Definition 7. *In $(\mathbb{G}, \mathbb{G}_T)$, the q-***Decision Bilinear Diffie-Hellman Exponent** *(q-DBDHE) problem is, given a tuple $(g, g^\gamma, g^{(\gamma^2)}, \ldots, g^{(\gamma^q)}, g^{(\gamma^{q+2})}, \ldots, g^{(\gamma^{2q})}, h, T)$ where $\gamma \xleftarrow{R} \mathbb{Z}_p$, $g, h \xleftarrow{R} \mathbb{G}$ and $T \in_R \mathbb{G}_T$, to decide if $T = e(g, h)^{(\gamma^{q+1})}$ or if T is a random element of $\mathbb{G}_T$.*

3 Monotonic KP-ABE with Short Ciphertexts

Our first goal is to construct monotonic KP-ABE with short ciphertexts. We do so by showing a general transformation that automatically turns any IBBE scheme fitting a certain template into a KP-ABE in the selective security model.

The construction is somewhat similar to the one described by Boyen [12], which transforms IBE in the exponent-inversion framework to ABE. The approach of [12] took advantage of certain linearity properties in a family of IBE schemes. Our approach also exploits some linearity properties, albeit instead of IBE, we use IBBE as the underlying primitive. In contrast to [12], our transformation preserves the ciphertext size, hence using IBBE with short ciphertexts will yield KP-ABE with the same ciphertext size.

3.1 Linear ID-based Broadcast Encryption Template

We define a template that IBBE schemes should comply with in order to give rise to (selectively secure) KP-ABE schemes. We call this a linear IBBE template. Let $(\mathbb{G}, \mathbb{G}_T)$ be underlying bilinear groups of order p. A linear IBBE scheme is determined by parameter $n_1, n_2 \in \mathbb{N}$, a family $\mathcal{F}$ of vectors of functions, and a function $\mathcal{D}$, of which the latter two are specified by

$$\mathcal{F} \subset \big\{(f_1, f_2, F) \mid f_1 : \mathbb{Z}_p^* \to \mathbb{G},\ f_2 : \mathbb{Z}_p^* \to \mathbb{G}^{n_1},\ F : (\mathbb{Z}_p^*)^{\le n-1} \to \mathbb{G}^{\le n_2}\big\},$$
$$\mathcal{D} : \mathbb{G}^{n_1+2} \times \mathcal{I} \times \mathbb{G}^{\le n_2+1} \times \tbinom{\mathcal{I}}{<n} \to \mathbb{G}_T,$$

with requirements specified below. A linear IBBE scheme works as follows.

▶ $\mathsf{Setup}(\lambda, n)$: Given a security parameter $\lambda \in \mathbb{N}$ and a bound $n \in \mathbb{N}$ on the number of identities per ciphertext, the algorithm selects bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p and a generators $g \xleftarrow{R} \mathbb{G}$. It computes $e(g,g)^\alpha$ for a random $\alpha \xleftarrow{R} \mathbb{Z}_p^*$ and chooses functions $(f_1, f_2, F) \xleftarrow{R} \mathcal{F}$. The master secret key consists of $\mathsf{msk} := g^\alpha$ while the public key is $\mathsf{mpk} := \big(g,\ e(g,g)^\alpha,\ f_1,\ f_2,\ F,\ n,\ n_1,\ n_2\big)$.

▶ $\mathsf{Keygen}(\mathsf{msk}, \mathsf{ID})$: It picks $r \xleftarrow{R} \mathbb{Z}_p^*$ and computes

$$\mathsf{sk}_{\mathsf{ID}} = (d_1, d_2, d_3) = \big(g^\alpha \cdot f_1(\mathsf{ID})^r,\ g^r,\ f_2(\mathsf{ID})^r\big) \in \mathbb{G}^{n_1+2}.$$

▶ $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, S)$: It parses S as $S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_q\}$, where $q < n$. To encrypt $\mathsf{M} \in \mathbb{G}_T$, it chooses a random exponent $s \xleftarrow{R} \mathbb{Z}_p^*$ and computes the ciphertext as

$$C = (C_0, C_1, C_2) = \Big(\mathsf{M} \cdot e(g,g)^{\alpha \cdot s},\ g^s,\ F(\mathsf{ID}_1, \ldots, \mathsf{ID}_q)^s\Big).$$

▶ $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{ID}, C, S)$: It parses $\mathsf{sk}_{\mathsf{ID}} = (d_1, d_2, d_3)$ and $C = (C_0, C_1, C_2)$ then runs

$$\mathcal{D}\big((d_1, d_2, d_3), \mathsf{ID}, (C_1, C_2), S\big) \to e(g,g)^{\alpha \cdot s},$$

and obtains $\mathsf{M} = C_0 / e(g,g)^{\alpha \cdot s}$. We are now ready to state the requirements: for all $(f_1, f_2, F) \in \mathcal{F}$, the following two properties must hold.

1. **Correctness.** For all $\alpha, r, s \in \mathbb{Z}_p^*$, $\mathsf{ID} \in \mathcal{I}$, $S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_q\} \in \binom{\mathcal{I}}{<n}$ and $\mathsf{ID} \in S$, we have

$$\mathcal{D}\Big(\big(g^\alpha f_1(\mathsf{ID})^r, g^r, f_2(\mathsf{ID})^r\big), \mathsf{ID}, (g^s, F(\mathsf{ID}_1, \ldots, \mathsf{ID}_q)^s), S\Big) = e(g,g)^{\alpha \cdot s}.$$

2. **Linearity.** For all $\gamma \in \mathbb{Z}_p^*$, $\mathsf{ID} \in \mathcal{I}$, $S \in \binom{\mathcal{I}}{<n}$, $\mathsf{ID} \in S$, all keys $(d_1, d_2, d_3) \in \mathbb{G}^{n_1+2}$ and all $(C_1, C_2) \in \mathbb{G}^{\le n_2+1}$, we have

$$\mathcal{D}\Big((d_1, d_2, d_3)^\gamma, \mathsf{ID}, (C_1, C_2), S\Big) = \mathcal{D}\Big((d_1, d_2, d_3), \mathsf{ID}, (C_1, C_2), S\Big)^\gamma.$$

3.2 Generic Conversion from Linear IBBE to KP-ABE

Let $\Pi_{\mathsf{IBBE}} = (\mathsf{Setup}', \mathsf{Keygen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ be a linear IBBE system. We construct a KP-ABE scheme from Π_{IBBE} as follows.

▶ $\mathsf{Setup}(\lambda, n)$: It simply outputs $\mathsf{Setup}'(\lambda, n) \to (\mathsf{msk}, \mathsf{mpk})$.
▶ $\mathsf{Keygen}(\mathsf{msk}, (L, \pi))$: The algorithm computes a private key for an access structure that is associated with LSSS scheme (L, π) as follows. Let L be $\ell \times k$ matrix. First, it generates shares of 1 with the LSSS (L, π). Namely, it chooses a vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)^\top \xleftarrow{R} (\mathbb{Z}_p)^k$ subject to the constraint $\beta_1 = 1$. Then for each $i = 1$ to ℓ, it calculates $\lambda_i = \langle \boldsymbol{L_i}, \boldsymbol{\beta} \rangle$, picks $r' \xleftarrow{R} \mathbb{Z}_p$ and sets D_i as follows.

$$\begin{aligned} &\mathsf{Keygen}'(\mathsf{msk}, \pi(i)) \to (d_{i,1},\ d_{i,2},\ d_{i,3}), \\ &\qquad D_i = \big(d_{i,1}^{\lambda_i} \cdot f_1(\pi(i))^{r'},\ d_{i,2}^{\lambda_i} \cdot g^{r'},\ d_{i,3}^{\lambda_i} \cdot f_2(\pi(i))^{r'}\big). \end{aligned}$$

It then outputs the private key as $\mathsf{sk}_{(L,\pi)} = \{D_i\}_{i=1,\ldots,\ell}$.
▶ $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \omega)$: It simply outputs $\mathsf{Encrypt}'(\mathsf{mpk}, \mathsf{M}, \omega) \to (C_0, C_1, C_2)$.
▶ $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{(L,\pi)}, (L, \pi), C, \omega)$: Assume first that the policy (L, π) is satisfied by the attribute set ω, so that decryption is possible. Let $I = \{i|\ \pi(i) \in \omega\}$. It calculates the reconstruction constants $\{(i, \mu_i)\}_{i \in I} = \mathsf{Recon}_{(L,\pi)}(\omega)$. It parses C as (C_0, C_1, C_2) and $\mathsf{sk}_{(L,\pi)}$ as $\{D_i\}_{i=1,\ldots,\ell}$ where $D_i = (d'_{i,1}, d'_{i,2}, d'_{i,3})$. For each $i \in I$, it computes

$$\mathcal{D}\big((d'_{i,1}, d'_{i,2}, d'_{i,3}), \mathsf{ID}, (C_1, C_2), S\big) \to e(g, g)^{\alpha \cdot s \cdot \lambda_i}, \tag{1}$$

which we prove correctness below. It computes $e(g, g)^{\alpha \cdot s} = \prod_{i \in I} \big(e(g, g)^{\alpha \cdot s \cdot \lambda_i}\big)^{\mu_i}$ and finally obtains $\mathsf{M} = C_0 / e(g, g)^{\alpha \cdot s}$, where we recall that $\sum_{i \in I} \lambda_i \mu_i = 1$.

Correctness. We now verify that equation (1) is correct. First from a property of keys in linear IBBE, we have that $(d_{i,1}, d_{i,2}, d_{i,3})$ will be in the form $\big(g^\alpha \cdot f_1(\pi(i))^{r_i}, g^{r_i}, f_2(\pi(i))^{r_i}\big)$ for some $r_i \in_R \mathbb{Z}_p$. Therefore, we have

$$D_i = \big(g^{\alpha \lambda_i} \cdot f_1(\pi(i))^{\tilde{r}_i \lambda_i}, g^{\tilde{r}_i \lambda_i}, f_2(\pi(i))^{\tilde{r}_i \lambda_i}\big) = \big(d_1^{\lambda_i}, d_2^{\lambda_i}, d_3^{\lambda_i}\big),$$

with $\tilde{r}_i = r_i + r'/\lambda_i$ and $(d_1, d_2, d_3) = \mathsf{sk}_{\pi(i)}$ with randomness $\tilde{r}_i$. Hence,

$$\begin{aligned} \mathcal{D}\big((d'_{i,1}, d'_{i,2}, d'_{i,3}), \mathsf{ID}, (C_1, C_2), S\big) &= \mathcal{D}\big((d_1, d_2, d_3), \mathsf{ID}, (C_1, C_2), S\big)^{\lambda_i} \\ &= \big(e(g, g)^{\alpha \cdot s}\big)^{\lambda_i}, \end{aligned}$$

where each equality holds from linearity and correctness of $\mathcal{D}$ respectively.

The construction only guarantees selective security for the resulting KP-ABE. It does not extend to the adaptive scenario because the proof relies on the fact that the reduction knows the forbidden attribute set from the beginning.

Theorem 1. *If the underlying IBBE scheme is selectively secure, then the resulting KP-ABE system is also selectively secure.* (The proof is given in appendix B).

Instantiation Example. The large-universe construction of KP-ABE in [20] falls into our framework here. Its underlying IBBE system can be seen as a particular instance of the linear IBBE template with $n_2 = n$, $f_2(\mathsf{ID}) = \emptyset$, $F(\mathsf{ID}_1, \ldots, \mathsf{ID}_q) = (f_1(\mathsf{ID}_1), \ldots, f_1(\mathsf{ID}_q))$, and the form of f_1 can be straightforwardly deduced from [20]. Since the size of an output from F is linear, ciphertexts in the KP-ABE of [20] are also of linear size.

3.3 IBBE Instantiation with Short Ciphertexts

This subsection presents an IBBE scheme with short ciphertexts and shows how to apply the KP-ABE conversion. This specific IBBE can be seen as an instance of the functional encryption (FE) for zero inner-product proposed in [4, Sect.4.1], which itself is implied by spatial encryption of [10]. A FE system for zero inner-product is defined by a relation $R^{\mathsf{ZIP}} : \mathbb{Z}_p \times \mathbb{Z}_p \to \{0,1\}$ where $R^{\mathsf{ZIP}}(\boldsymbol{X},\boldsymbol{Y}) = 1$ iff $\langle \boldsymbol{X},\boldsymbol{Y}\rangle = 0$. The technique of deriving an IBBE scheme from a FE scheme for zero inner-product can be traced to [23]. A private key for an identity ID is defined by setting $\boldsymbol{X} = (x_1,\ldots,x_n)^\top$, with $x_i = \mathsf{ID}^{i-1}$. To encrypt to a set $S = \{\mathsf{ID}_1,\ldots,\mathsf{ID}_q\}$, one defines $\boldsymbol{Y} = (y_1,\ldots,y_n)^\top$ as a coefficient vector from

$$P_S[Z] = \sum_{i=1}^{q+1} y_i Z^{i-1} = \prod_{\mathsf{ID}_j \in S} (Z - \mathsf{ID}_j), \tag{2}$$

where, if $q+1 < n$, the coordinates $y_{q+2},\ldots,y_n$ are all set to 0. By doing so, we note that $P_S[\mathsf{ID}] = \langle \boldsymbol{X},\boldsymbol{Y}\rangle$ evaluates to 0 iff $\mathsf{ID} \in S$. We now describe the IBBE instantiated from the FE system of [4]. Its selective security is an immediate consequence of [4], where it is proved under the DBDHE assumption.

▶ $\mathsf{Setup}(\lambda, n)$: It chooses bilinear groups $(\mathbb{G},\mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{R} \mathbb{G}$. It randomly chooses $\alpha,\alpha_0 \xleftarrow{R} \mathbb{Z}_p$, $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_n)^\top \xleftarrow{R} \mathbb{Z}_p^n$. It then sets $\boldsymbol{H} = (h_1,\ldots,h_n)^\top = g^{\boldsymbol{\alpha}}$. The master secret key is $\mathsf{msk} = \alpha$, and the public key is $\mathsf{mpk} = \big(g,\ e(g,g)^\alpha,\ h_0 = g^{\alpha_0},\ \boldsymbol{H} = g^{\boldsymbol{\alpha}}\big)$.

▶ $\mathsf{Keygen}(\mathsf{msk},\mathsf{ID})$: The algorithm first defines a vector $\boldsymbol{X} = (x_1,\ldots,x_n)^\top$ such that $x_i = \mathsf{ID}^{i-1}$ for $i = 1$ to n. It chooses $r \xleftarrow{R} \mathbb{Z}_p$ and outputs the private key as $\mathsf{sk}_{\mathsf{ID}} = (D_1, D_2, K_2,\ldots,K_n)$ where

$$D_1 = g^\alpha \cdot h_0^r, \qquad D_2 = g^r, \qquad \Big\{K_i = \big(h_1^{-\frac{x_i}{x_1}} \cdot h_i\big)^r\Big\}_{i=2,\ldots,n}.$$

▶ $\mathsf{Encrypt}(\mathsf{mpk},\mathsf{M},S)$: To encrypt M to the receiver set S (where $|S| < n$), the algorithm defines $\boldsymbol{Y} = (y_1,\ldots,y_n)^\top$ as the coefficient vector of $P_S[Z]$ from equation (2). It then picks $s \xleftarrow{R} \mathbb{Z}_p$ and computes the ciphertext as

$$C = (C_0, C_1, C_2) = \Big(\mathsf{M} \cdot e(g,g)^{\alpha s},\ g^s,\ \big(h_0 \cdot h_1^{y_1} \cdots h_n^{y_n}\big)^s\Big).$$

▶ $\mathsf{Decrypt}(\mathsf{mpk},\mathsf{sk}_{\mathsf{ID}},\mathsf{ID},C,S)$: It defines the vector $\boldsymbol{Y} = (y_1,\ldots,y_n)^\top$ from the polynomial $P_S[Z]$ as usual. It then computes

$$e(g,g)^{\alpha\cdot s} = \frac{e(C_1, D_1 \cdot K_2^{y_2} \cdots K_n^{y_n})}{e(C_2, D_2)}, \tag{3}$$

and recovers $\mathsf{M} = C_0/e(g,g)^{\alpha\cdot s}$.

Correctness. If $\langle \boldsymbol{X},\boldsymbol{Y}\rangle = 0$, then decryption recovers M since

$$D_1 \cdot \prod_{i=2}^n K_i^{y_i} = g^\alpha \cdot \Big(h_0 \cdot h_1^{-\frac{1}{x_1}(\langle \boldsymbol{X},\boldsymbol{Y}\rangle - x_1 y_1)} \prod_{i=2}^n h_i^{y_i}\Big)^r = g^\alpha \cdot \Big(h_0 \cdot \prod_{i=1}^n h_i^{y_i}\Big)^r,$$

so that $e(C_1, D_1 \cdot \prod_{i=1}^n K_i^{y_i}) = e(g,g)^{\alpha s} \cdot e(h_0 \cdot \prod_{i=1}^n h_i^{y_i}, g^{rs})$ equals the product $e(g,g)^{\alpha s} \cdot e(C_2, D_2)$.

Applying the KP-ABE Conversion. The above IBBE can be considered as a linear IBBE system with $n_1 = n - 1$, $n_2 = 1$ and the family $\mathcal{F}$ is defined by taking all functions of the following forms ranging over $h_0, h_1, \ldots, h_n \in \mathbb{G}$:

$$f_1(\mathsf{ID}) = h_0,\ f_2(\mathsf{ID}) = \big(h_1^{-\mathsf{ID}} h_2, \ldots, h_1^{-\mathsf{ID}^{n-1}} h_n\big),\ F(\mathsf{ID}_1, \ldots, \mathsf{ID}_q) = h_0 \prod_{i=1}^{q+1} h_i^{y_i},$$

where the vector $\boldsymbol{Y} = (y_1, \ldots, y_n)^\top$ is defined from the polynomial $P_S[Z]$ in equation (2) as usual. In addition, the function $\mathcal{D}$ is the computation in equation (3), which can be shown to have linearity as required.

The resulting KP-ABE has constant-size ciphertexts. This comes with the expense of longer private keys of size $O(t \cdot n)$, where t is the number of attributes in the access structure. It is also worth mentioning that we can obtain another IBBE with short ciphertexts from the spatial encryption scheme of [10] since it also falls into our framework and thus produces another KP-ABE scheme.

Our goal in this paper is to construct KP-ABE with non-monotonic structures. We will combine the monotonic KP-ABE system in this subsection with new ID-based revocation in the next section.

4 Revocation Scheme with Very Short Ciphertexts

This section describes a new ID-based revocation system which is tailored to the needs of our application. Analogously to the case of IBBE, an IBR scheme can be instantiated from a FE system for *non-zero* inner-product relations. Two such existing IBR schemes [4, Sect. 5.1 and 5.2] already provide constant-size ciphertexts. When it comes to construct a non-monotonic KP-ABE however, these schemes seem hardly compatible with the monotonic KP-ABE of section 3.3 as they rely on different assumptions. We thus describe a new IBR scheme for this purpose. Its structure is similar to that of revocation schemes given in [4] but it provides a better efficiency and relies on the DBDHE assumption.

▶ $\mathsf{Setup}(\lambda, n)$: It chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ and a generator $g \overset{R}{\leftarrow} \mathbb{G}$. It randomly picks $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \overset{R}{\leftarrow} \mathbb{Z}_p^n$ and sets $\boldsymbol{H} = (h_1, \ldots, h_n)^\top = g^{\boldsymbol{\alpha}}$. The master secret key is $\mathsf{msk} = \alpha$, while the public key is $\mathsf{mpk} = \big(g,\ e(g,g)^\alpha,\ \boldsymbol{H} = g^{\boldsymbol{\alpha}}\big)$.

▶ $\mathsf{Keygen}(\mathsf{msk}, \mathsf{ID})$: The algorithm first defines a vector $\boldsymbol{X} = (x_1, \ldots, x_n)^\top$ such that $x_i = \mathsf{ID}^{i-1}$ for $i = 1$ to n. It chooses $r \overset{R}{\leftarrow} \mathbb{Z}_p$ and outputs the private key as $\mathsf{sk}_{\mathsf{ID}} = (D_1, D_2, K_2, \ldots, K_n)$ where

$$D_1 = g^\alpha \cdot h_1^r, \qquad D_2 = g^r, \qquad \Big\{K_i = \big(h_1^{-\frac{x_i}{x_1}} \cdot h_i\big)^r\Big\}_{i=2,\ldots,n}.$$

Indeed, we can also write $K_{\boldsymbol{X}} = (K_2, \ldots, K_n) = g^{r \cdot M_{\boldsymbol{X}}^\top \boldsymbol{\alpha}}$, where the matrix $M_{\boldsymbol{X}} \in (\mathbb{Z}_p)^{n \times (n-1)}$ is defined by $M_{\boldsymbol{X}} = \begin{pmatrix} -\frac{x_2}{x_1} & -\frac{x_3}{x_1} & \cdots & -\frac{x_n}{x_1} \\ & I_{n-1} & & \end{pmatrix}$.

▶ $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, S)$: To encrypt M with the revoked set S (where $|S| < n$), the algorithm defines $\boldsymbol{Y} = (y_1, \ldots, y_n)^\top$ as the coefficient vector of $P_S[Z]$ from equation (2). It then picks $s \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes the ciphertext as

$$C = (C_0, C_1, C_2) = \Big(\mathsf{M} \cdot e(g,g)^{\alpha \cdot s},\ g^s,\ \big(h_1^{y_1} \cdots h_n^{y_n}\big)^s\Big).$$

▶ $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{ID}, C, S)$: It defines $\boldsymbol{X}$ from ID and $\boldsymbol{Y}$ from S as usual. It then successively computes elements $K = \prod_{i=2}^{n} K_i^{y_i} = \big(h_1^{-\langle \boldsymbol{X},\boldsymbol{Y}\rangle/x_1} \cdot h_1^{y_1} \cdots h_n^{y_n}\big)^r$, $\tau = \Big(\frac{e\left(K,C_1\right)}{e(C_2,D_2)}\Big)^{-\frac{x_1}{\langle \boldsymbol{X},\boldsymbol{Y}\rangle}} = e(g,h_1)^{rs}$, and then obtains $\mathsf{M} = C_0 \cdot e(C_1, D_1)^{-1} \cdot \tau$.

Correctness. We first observe that

$$K = \big(h_1^{-(\langle \boldsymbol{X},\boldsymbol{Y}\rangle - x_1 y_1)/x_1} \prod_{i=2}^{n} h_i^{y_i}\big)^r = \big(h_1^{-\langle \boldsymbol{X},\boldsymbol{Y}\rangle/x_1} \prod_{i=1}^{n} h_i^{y_i}\big)^r$$

so that whenever $\langle \boldsymbol{X},\boldsymbol{Y}\rangle \neq 0$ (*i.e.*, $\mathsf{ID} \notin S$), the following computation can be done.

$$\tau = \Big(\frac{e(K,C_1)}{e(C_2,D_2)}\Big)^{-\frac{x_1}{\langle \boldsymbol{X},\boldsymbol{Y}\rangle}} = \Big(\frac{e(h_1^{-\langle \boldsymbol{X},\boldsymbol{Y}\rangle/x_1} \prod_{i=1}^{n} h_i^{y_i}, g^{rs})}{e(\prod_{i=1}^{n} h_i^{y_i}, g^{rs})}\Big)^{-\frac{x_1}{\langle \boldsymbol{X},\boldsymbol{Y}\rangle}} = e(g,h_1)^{rs}.$$

Finally, we have $e(C_1, D_1) \cdot \tau^{-1} = e(g,g)^{\alpha \cdot s} \cdot e(g^s, h_1^r) \cdot e(g,h_1)^{-rs} = e(g,g)^{\alpha \cdot s}$. We note that the decryption algorithm can be optimized by computing the plaintext as

$$\mathsf{M} = C_0 \cdot e\big(C_2, D_2^{x_1/\langle \boldsymbol{X},\boldsymbol{Y}\rangle}\big) \cdot e\big(C_1, D_1^{-1} \cdot K^{-x_1/\langle \boldsymbol{X},\boldsymbol{Y}\rangle}\big).$$

At a high level, it shares the same structure (including the form of the public key and the ciphertext) as the IBBE in section 3.3 and relies on the same assumption. Intuitively, these similarities make it possible to assemble both constructions in the design of a non-monotonic ABE system in section 5.

We now prove the co-selective security of the scheme. It is also worth recalling that co-selective security for IBR also implies selective security.

Theorem 2. *The above ID-based revocation scheme with the maximal bound n for the number of revoked users (i.e., $|S| < n$) is co-selectively secure if the n-DBDHE assumption holds in $(\mathbb{G}, \mathbb{G}_T)$.*

Proof. We show an algorithm $\mathcal{B}$ that receives $(g, h, z_1, \ldots, z_n, z_{n+2}, \ldots, z_{2n}, T)$ in $\mathbb{G}^{2n+1} \times \mathbb{G}_T$, where $z_i = g^{(\gamma^i)}$, and decides if $T = e(g,h)^{(\gamma^{n+1})}$ using the co-selective adversary $\mathcal{A}$.

At the outset of the game, the adversary $\mathcal{A}$ declares the set $\tilde{S} = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_q\}$, where $q \leq n-1$, of identities for which she wishes to obtain private keys. Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_q$ the corresponding vectors. That is, $\boldsymbol{X}_k = (1, \mathsf{ID}_k, \mathsf{ID}_k^2, \ldots, \mathsf{ID}_k^{n-1})$. To prepare the public key, $\mathcal{B}$ chooses $\delta_0 \xleftarrow{R} \mathbb{Z}_p$ and computes $e(g,g)^{\alpha} = e(z_1, z_n)^{\delta_0}$, which implicitly defines $\alpha = \gamma^{(n+1)} \cdot \delta_0$. Elements $\boldsymbol{H} = (h_1, \ldots, h_n)^{\top}$ are then defined as follows. For each $k \in [1,q]$, $\mathcal{B}$ considers the vector $\boldsymbol{X}_k = (x_{k,1}, \ldots, x_{k,n})^{\top}$ and selects $\boldsymbol{b}_k \in \mathbb{Z}_p^n$ such that

$$\boldsymbol{b}_k{}^{\top} \cdot M_{\boldsymbol{X_k}} = \boldsymbol{b}_k{}^{\top} \cdot \begin{pmatrix} -\frac{x_{k,2}}{x_{k,1}} & -\frac{x_{k,3}}{x_{k,1}} & \cdots & -\frac{x_{k,n}}{x_{k,1}} \\ & I_{n-1} & & \end{pmatrix} = \boldsymbol{0}. \qquad (4)$$

The simplest candidate consists of the vector $\boldsymbol{b}_k = (1, \frac{x_{k,2}}{x_{k,1}}, \frac{x_{k,3}}{x_{k,1}}, \ldots, \frac{x_{k,n}}{x_{k,1}})^{\top}$. Then, $\mathcal{B}$ considers the $n \times n$ matrix $B = \big(\boldsymbol{b}_1 | \ldots | \boldsymbol{b}_q | \boldsymbol{0} | \ldots | \boldsymbol{0}\big)$ whose k^{th} column consists of $\boldsymbol{b}_k$, for $k = 1$ to q, and where the $n - q$ remaining columns are $\boldsymbol{0}$. It defines $\boldsymbol{a} = (a_1, \ldots, a_n)^{\top} \in (\mathbb{Z}_p)^n$ such that $a_i = \gamma^{n+1-i}$ by setting $g^{\boldsymbol{a}} = (z_n, \ldots, z_1)^{\top}$. Then, it implicitly sets $\boldsymbol{\alpha} = B \cdot \boldsymbol{a} + \boldsymbol{\delta}$ by randomly choosing $\boldsymbol{\delta} \xleftarrow{R} \mathbb{Z}_p^n$ and defining $\boldsymbol{H} = g^{B \cdot \boldsymbol{a}} \cdot g^{\boldsymbol{\delta}}$, which is uniformly distributed as required.

Due to (4), the matrix B is defined in such a way that, for each $k \in [1,q]$, the k^{th} column of

$M_{\boldsymbol{X}_k}^\top \cdot B \in (\mathbb{Z}_p)^{(n-1)\times n}$ is $\mathbf{0}$, so that $M_{\boldsymbol{X}_k}^\top \cdot B \cdot \boldsymbol{a}$ does not contain $a_k = \gamma^{n+1-k}$. Then, a private key for the identity ID_k (and thus the vector $\boldsymbol{X}_k$) can be obtained by implicity defining $\tilde{r}_k = r_k - \delta_0 \gamma^k$ for a random $r_k \xleftarrow{R} \mathbb{Z}_p$. Indeed, with the above choice of B, the first coordinate of $\boldsymbol{\alpha} = \boldsymbol{\delta} + \sum_{j=1}^q a_j \boldsymbol{b}_j$ equals $\alpha_1 = \delta_1 + \sum_{j=1}^q a_j = \delta_1 + \sum_{j=1}^q \gamma^{(n+1-j)}$, so that $\mathcal{B}$ is able to compute

$$\begin{aligned} D_1 = g^\alpha \cdot h_1^{\tilde{r}_k} &= g^{(\gamma^{n+1})\delta_0} \cdot h_1^{r_k} \cdot \Big(g^{\delta_1} \cdot \prod_{j=1}^q z_{n+1-j} \Big)^{-\delta_0 \gamma_k} \\ &= h_1^{r_k} \cdot \Big(z_k^{\delta_1} \cdot \prod_{j=1, j\neq k}^q z_{n+1-j+k} \Big)^{-\delta_0} \end{aligned}$$

and $D_2 = g^{r_k} \cdot z_{n+1-k}^{-\delta_0}$. As for the delegation component $K_{\boldsymbol{X}_k} = g^{\tilde{r}_k M_{\boldsymbol{X}_k}^\top \boldsymbol{\alpha}}$, $\mathcal{B}$ is also able to compute it from available values since $M_{\boldsymbol{X}_k}^\top \boldsymbol{\alpha} = M_{\boldsymbol{X}_k}^\top \cdot B \cdot \boldsymbol{a} + M_{\boldsymbol{X}_k}^\top \cdot \boldsymbol{\delta}$ is independent of $a_k = \gamma^{n+1-k}$ (recall that the k^{th} column of $M_{\boldsymbol{X}_k}^\top \cdot B$ is $\mathbf{0}$) and no term γ^{n+1} appears in the exponent in $K_{\boldsymbol{X}_k}$.

In the challenge phase, $\mathcal{B}$ chooses $\mathsf{M}_0, \mathsf{M}_1 \in \mathbb{G}_T$ and a revocation set S corresponding to a vector $\boldsymbol{Y} = (y_1, \ldots, y_n)^\top$ that must satisfy $\langle \boldsymbol{X}_k, \boldsymbol{Y} \rangle = 0$ for $k = 1$ to q. This amounts to say that $\boldsymbol{Y} = M_{\boldsymbol{X}_k} \cdot \boldsymbol{w}$, where $\boldsymbol{w} = (y_2, \ldots, y_n)^\top$, for each $k \in [1, q]$ (see equations (6)-(7) in appendix C for a justification of this statement). We claim that $\boldsymbol{Y}^\top \cdot B \cdot \boldsymbol{a} = 0$. Indeed,

$$\boldsymbol{Y}^\top \cdot B \cdot \boldsymbol{a} = \boldsymbol{Y}^\top \cdot \Big(\sum_{k=1}^q a_k \cdot \boldsymbol{b}_k \Big) = \sum_{k=1}^q a_k \cdot \boldsymbol{Y}^\top \cdot \boldsymbol{b}_k = \sum_{k=1}^q a_k \cdot \boldsymbol{w}^\top \cdot M_{\boldsymbol{X}_k}^\top \cdot \boldsymbol{b}_k$$

and $M_{\boldsymbol{X}_k}^\top \cdot \boldsymbol{b}_k = \mathbf{0}$ for each $k \in [1, q]$. Therefore, it comes that $\langle \boldsymbol{Y}, \boldsymbol{\alpha} \rangle = \langle \boldsymbol{Y}, \boldsymbol{\delta} \rangle$, so that $\mathcal{B}$ can generate a challenge ciphertext (C_0, C_1, C_2) as

$$C_0 = \mathsf{M}_d \cdot T^{\delta_0}, \qquad C_1 = h, \qquad C_2 = h^{\langle \boldsymbol{Y}, \boldsymbol{\delta} \rangle},$$

for a random bit $d \xleftarrow{R} \{0, 1\}$. If $T = e(g, h)^{(\gamma^{n+1})}$, $C = (C_0, C_1, C_2)$ forms a valid encryption of M_d. If T is random, C carries no information on $d \in \{0, 1\}$ and $\mathcal{A}$'s advantage is clearly zero. □

In the proof of the above theorem, we note that terms z_1 and z_{2n} are not used in the reduction. However, they will be used in the security proof of our non-monotonic ABE (where the reduction will set up part of the public parameters in a similar way to the proof of theorem 2) in section 5 and we thus used the n-DBDHE assumption for clarity.

Efficiency Comparisons. We believe this IBR scheme to be of interest in its own right. If we compare it with the scheme of [4, Sect.5.2] (called AL2 here), which also features short ciphertexts, it relies on a stronger assumption (since no "q-type" assumption is needed in [4] or in LSW2 [24]) but provides significantly shorter ciphertexts (as the ciphertext overhead is decreased by more than 75%)[7] and requires fewer pairing evaluations to decrypt (only 2 instead of 9). Another IBR scheme (dubbed AL1 in the table) with a better efficiency than AL2 was described in [4, Sect.5.1]. Still, the new scheme is slightly more efficient and relies on a weaker assumption since q-DBDHE is weaker and appears more natural than the q-type assumption (MEBDH) used in [24, 4].

In comparison with the schemes of Lewko, Sahai and Waters, the disadvantage lies in that a bound on the number of revocations must be chosen when the system is set up. A comparative efficiency of known IBR schemes is given in the table hereafter.

[7] We compare by simple element counting. In a stricter sense, one may want to also consider the compensation due to the attack on q-type assumptions by Cheon [17].

Table 1. Performances of revocation systems

Schemes	Ciphertext overhead	Private key size	Decryption cost		Assumption
	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}\rvert$	pair.	exp.	
LSW1 [24]	$(2\bar{n}+1)$	3	3	$O(\bar{n})$	n-MEBDH
LSW2 [24]	$(2\bar{n}+7)$	7	9	$O(\bar{n})$	DLIN, DBDH
AL1 [4]	3	$(n+2)$	3	$O(n)$	n-MEBDH
AL2 [4]	9	$(n+2)$	9	$O(n)$	DLIN, DBDH
This work	2	$(n+2)$	2	$O(n)$	n-DBDHE

† $\bar{n}$ = # of revoked users = $|S|$; n = the maximal bound for $\bar{n}$. (*i.e.*, $|S| < n$).
‡ pair.,exp. shows # of pairing and exponentiation computation.

5 Non-Monotonic KP-ABE with Short Ciphertexts

Ostrovsky, Sahai and Waters [28] suggested a technique to move from monotonic to non-monotonic access structures without incurring an immoderate private key size. They assume a family $\{\Pi_{\mathbb{A}}\}_{\mathbb{A}\in\mathcal{AS}}$ of linear secret-sharing schemes for a set of monotone access structures $\mathbb{A}$. For each such access structure $\mathbb{A} \in \mathcal{AS}$, the set $\mathcal{P}$ of underlying parties is defined in such a way that parties' names can be normal (like x) or primed (like x'). Prime attributes are conceptually seen as the negation of unprimed attributes. In addition, it is required that, if $x \in \mathcal{P}$, then $x' \in \mathcal{P}$ and vice versa.

A family $\mathcal{AS}$ of non-monotone access structures can be defined as follows. For each access structure $\mathbb{A} \in \mathcal{AS}$ over a set of parties $\mathcal{P}$, one defines a possibly non-monotonic access structure $NM(\mathbb{A})$ over the set $\tilde{\mathcal{P}}$ of all unprimed parties in $\mathcal{P}$. An operator $N(.)$ is then defined as follows. For every set $\tilde{S} \subset \tilde{\mathcal{P}}$, one imposes $\tilde{S} \subset N(\tilde{S})$. Also, for each $x \in \tilde{\mathcal{P}}$ such that $x \notin \tilde{S}$, $x' \in N(\tilde{S})$. Finally, $NM(\mathbb{A})$ is defined by saying that $\tilde{S}$ is authorized in $NM(\mathbb{A})$ if and only if $N(\tilde{S})$ is authorized in $\mathbb{A}$ (so that $NM(\mathbb{A})$ has only unprimed parties in its access sets). For each access set $X \in NM(\mathbb{A})$, there is a set in $\mathbb{A}$ containing the elements in X and primed elements for each party not in X.

In [28], the above technique was combined with the Naor-Pinkas revocation method [27] to cope with non-monotonic access structures. Lewko, Sahai and Waters provided improvements using a revocation system with short keys [24] instead of [27]. In the following, we apply the same technique to our revocation mechanism and combine it with the monotonic KP-ABE derived from the IBBE scheme of section 3.3 in order to handle non-negated attributes.

▶ $\mathsf{Setup}(\lambda, n)$: Given a security parameter $\lambda \in \mathbb{N}$ and a bound $n \in \mathbb{N}$ of the number of attributes per ciphertext, it chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^{\lambda}$ and $g \xleftarrow{R} \mathbb{G}$. It defines $\boldsymbol{H} = (h_1, \ldots, h_n)^{\top}$ and $\boldsymbol{U} = (u_0, \ldots, u_n)^{\top}$ such that $h_i = g^{\alpha_i}$, $u_j = g^{\beta_j}$ for each $i \in \{1, \ldots, n\}$ and $j \in \{0, \ldots, n\}$ where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^{\top} \xleftarrow{R} \mathbb{Z}_p^n$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_n)^{\top} \xleftarrow{R} \mathbb{Z}_p^{n+1}$. It then picks $\alpha \xleftarrow{R} \mathbb{Z}_p^*$ and computes $e(g,g)^{\alpha}$. The master secret key is $\mathsf{msk} = \alpha$ and the master public key is

$$\mathsf{mpk} = \big(g,\ e(g,g)^{\alpha},\ \boldsymbol{H} = g^{\boldsymbol{\alpha}},\ \boldsymbol{U} = g^{\boldsymbol{\beta}}\big).$$

▶ $\mathsf{Keygen}(\mathsf{msk}, \tilde{\mathbb{A}})$: Given a non-monotonic access structure $\tilde{\mathbb{A}}$ such that we have $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$ over a set $\mathcal{P}$ of attributes and associated with a linear secret sharing scheme Π, the algorithm applies Π to obtain shares $\{\lambda_i\}$ of the master secret key α. The party corresponding to share λ_i is denoted by $\breve{x}_i \in \mathcal{P}$, where x_i is the underlying attribute, and can be primed (*i.e.*, negated) or unprimed (non-negated). For each i, the algorithm chooses $r_i \xleftarrow{R} \mathbb{Z}_p$, defines $\boldsymbol{\rho}_i = (\rho_{i,1}, \ldots, \rho_{i,n})^{\top} = (1, x_i, x_i^2, \ldots, x_i^{n-1})^{\top}$. That is $\rho_{i,j} = x_i^{j-1}$. Then, it does as follows.

- For each i such that $\breve{x}_i$ is an unprimed (*i.e.*, non-negated) attribute, the key generation algorithm computes a tuple $D_i = (D_{i,1}^{(1)}, D_{i,2}^{(2)}, K_{\boldsymbol{\rho}_i,i}^{(3)}) \in \mathbb{G}^{n+1}$ where the first two elements are of the form $(D_{i,1}^{(1)}, D_{i,2}^{(1)}) = \big(g^{\lambda_i} \cdot u_0^{r_i},\ g^{r_i}\big)$ and the third one is a tuple

$$K_{\boldsymbol{\rho}_i,i}^{(1)} = (K_{i,2}^{(1)}, \dots, K_{i,n}^{(1)}) = \Big(\big(u_1^{-\frac{\rho_{i,2}}{\rho_{i,1}}} \cdot u_2\big)^{r_i}, \dots,\ \big(u_1^{-\frac{\rho_{i,n}}{\rho_{i,1}}} \cdot u_n\big)^{r_i}\Big) = g^{r_i \cdot M_{\boldsymbol{\rho}_i}^\top \boldsymbol{\beta}},$$

  where $M_{\boldsymbol{\rho}_i} \in (\mathbb{Z}_p)^{n \times (n-1)}$ is the matrix $M_{\boldsymbol{\rho}_i} = \begin{pmatrix} -\frac{\rho_{i,2}}{\rho_{i,1}} & -\frac{\rho_{i,3}}{\rho_{i,1}} & \cdots & -\frac{\rho_{i,n}}{\rho_{i,1}} \\ & I_{n-1} & & \end{pmatrix}$.
- For each i such that $\breve{x}_i$ is primed (*i.e.*, negated), the key generation algorithm computes a tuple $D_i = (D_{i,1}^{(2)}, D_{i,2}^{(2)}, K_{\boldsymbol{\rho}_i,i}^{(2)}) \in \mathbb{G}^{n+1}$ where $(D_{i,1}^{(2)}, D_{i,2}^{(2)}) = \big(g^{\lambda_i} \cdot h_1^{r_i},\ g^{r_i}\big)$ and

$$K_{\boldsymbol{\rho}_i,i}^{(2)} = (K_{i,2}^{(2)}, \dots, K_{i,n}^{(2)}) = \Big(\big(h_1^{-\frac{\rho_{i,2}}{\rho_{i,1}}} \cdot h_2\big)^{r_i}, \dots,\ \big(h_1^{-\frac{\rho_{i,n}}{\rho_{i,1}}} \cdot h_n\big)^{r_i}\Big) = g^{r_i \cdot M_{\boldsymbol{\rho}_i}^\top \boldsymbol{\alpha}}.$$

The private key is $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{\breve{x}_i \in \mathcal{P}} \in \mathbb{G}^{\ell \times (n+1)}$.

▶ $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \omega)$: To encrypt $\mathsf{M} \in \mathbb{G}_T$ for a set ω (with $|\omega| < n$), the algorithm first defines $\boldsymbol{Y} = (y_1, \dots, y_n)^\top$ as the vector whose first $q+1$ coordinates are the coefficients of the polynomial $P_\omega[Z] = \sum_{i=1}^{q+1} y_i Z^{i-1} = \prod_{j \in \omega}(Z - j)$. If $q+1 < n$, set $y_j = 0$ for $q+2 \leq j \leq n$. Then, it randomly picks $s \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes

$$C = (C_0, C_1, C_2, C_3) = \Big(\mathsf{M} \cdot e(g,g)^{\alpha \cdot s},\ g^s,\ \big(u_0 \cdot \prod_{i=1}^{n} u_i^{y_i}\big)^s,\ \big(\prod_{i=1}^{n} h_i^{y_i}\big)^s\Big).$$

▶ $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\tilde{\mathbb{A}}}, \tilde{\mathbb{A}}, C, \omega)$: It parses C as $(C_0, C_1, C_2, C_3) \in \mathbb{G}_T \times \mathbb{G}^3$ and the private key $\mathsf{sk}_{\tilde{\mathbb{A}}}$ as $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{\breve{x}_i \in \mathcal{P}} \in \mathbb{G}^{\ell \times (n+1)}$. The algorithm outputs $\perp$ if $\omega \not\in \tilde{\mathbb{A}}$. Otherwise, since $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some access structure $\mathbb{A}$ associated with a linear secret sharing scheme $\mathit{\Pi}$, we have $\omega' = N(\omega) \in \mathbb{A}$ and we let $I = \{i : \breve{x}_i \in \omega'\}$. Since ω' is authorized in $\mathbb{A}$, the receiver can efficiently compute coefficients $\{\mu_i\}_{i \in I}$ such that $\sum_{i \in I} \mu_i \lambda_i = \alpha$ (although the shares are not known to the receiver). Let $\boldsymbol{Y} = (y_1, \dots, y_n)^\top$ be the vector containing the coefficients of $P_\omega[Z] = \prod_{j \in \omega}(Z - j) = \sum_{i=1}^{q+1} y_i Z^{i-1}$.

- For every positive attribute $\breve{x}_i \in \omega'$ (for which $x_i \in \omega$), the decryption procedure computes $\tilde{D}_{i,1}^{(1)} = D_{i,1}^{(1)} \cdot \prod_{j=2}^{n} K_{i,j}^{(1)\,y_j} = g^\alpha \cdot \big(u_0 \cdot u_1^{y_1} \cdots u_n^{y_n}\big)^{r_i}$, and then $e(g,g)^{\lambda_i s} = e(C_1, \tilde{D}_{i,1}^{(1)}) / e(C_2, D_{i,2}^{(1)})$.
- For each negated attribute $\breve{x}_i \in \omega'$ (for which $x_i \not\in \omega$), the receiver sets $\boldsymbol{\rho}_i = (1, x_i, \dots, x_i^{n-1})^\top$ and successively computes

$$K_i^{(2)} = \prod_{j=2}^{n} K_{i,j}^{(2)\,y_j} = \big(h_1^{-\langle \boldsymbol{\rho_i}, \boldsymbol{Y} \rangle / x_1} \cdot h_1^{y_1} \cdots h_n^{y_n}\big)^{r_i},$$

$$\tau_i = \Big(\frac{e\big(K_i^{(2)}, C_1\big)}{e(C_3, D_{i,2}^{(2)})}\Big)^{-\frac{\rho_{i,1}}{\langle \boldsymbol{\rho}_i, \boldsymbol{Y} \rangle}} = e(g, h_1)^{r_i s}$$

  and then $e(g,g)^{\lambda_i s} = e(C_1, D_{i,1}^{(2)})^{-1} \cdot \tau_i^{-1}$.

Finally, decryption computes $\mathsf{M} = C_0 \cdot \prod_{i \in I} e(g,g)^{-\mu_i \lambda_i s}$.

If we split I into $I_0 \cup I_1$, where I_0 and I_1 correspond to unprimed and primed attributes, respectively, decryption can more efficiently compute

$$e(g,g)^{\alpha\cdot s} = e\Big(C_1, \prod_{i\in I_0} \tilde{D}_{i,1}^{(1)^{\mu_i}} \cdot \prod_{i\in I_1} \big(D_{i,1}^{(2)} \cdot K_i^{(2)\,\frac{\mu_i\cdot\rho_{i,1}}{\langle\rho_i,\boldsymbol{Y}\rangle}}\big)\Big) \cdot e\Big(C_2, \prod_{i\in I_0} D_{i,2}^{(1)^{\mu_i}}\Big) \cdot e\Big(C_3, \prod_{i\in I_1} D_{i,2}^{(2)\,\frac{\mu_i\cdot\rho_{i,1}}{\langle\rho_i,\boldsymbol{Y}\rangle}}\Big),$$

so that only three pairing evaluations are necessary.

Theorem 3. *The above KP-ABE system with the maximal bound n for the number of attributes per ciphertext (i.e., $|\omega| < n$) is selectively secure if the n-DBDHE assumption holds.* (The proof is deferred to appendix D).

6 Comparisons

Table 2 compares efficiency among available expressive KP-ABE schemes that support non-monotonic access structures. Comparisons are made in terms of ciphertext overhead, private key size as well as in the number of pairing evaluations and exponentiations (in $\mathbb{G}$ and $\mathbb{G}_T$) upon decryption.

We remark that the functionality of KP-ABE in [31] is slightly different from the original one [20]. For self-containment, we re-formalize it in appendix A, where we also briefly propose a modification of KP-ABE [31] so as to have the same functionality as the original ABE. We also include this modified scheme in Table 2. Note that [31] has a unique feature of being adaptively secure.

Table 2. Efficiency of non-monotonic KP-ABE schemes

Schemes	Ciphertext overhead	Private key size	Decryption cost		Assumption
	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}\rvert$	pair.	exp.	
OSW [28]	$O(\bar{n})$	$O(t\cdot\log n)$	$O(t)$	$O(t\cdot\bar{n})$	DBDH
LSW [24]	$O(\bar{n})$	$O(t)$	$O(t)$	$O(t\cdot\bar{n})$	n-MEBDH
OT [31]	$O(\bar{n}\cdot\varphi)$	$O(t\cdot\varphi)$	$O(t\cdot\varphi)$	$O(t)$	DLIN
$\text{OT}^{\text{modified}}$	$O(\bar{n}\cdot n)$	$O(t\cdot n)$	$O(t\cdot n)$	$O(t)$	DLIN
This work	3	$O(t\cdot n)$	3	$O(t)$	n-DBDHE

† $\bar{n}$ = |attribute set| = $|\omega|$ for a ciphertext; n =the maximal bound for $\bar{n}$ (*i.e.,* $|\omega| < n$); t = # of attributes in an access structure for a key; φ =maximum size for repetition of attribute label per key (only for the KP-ABE with labeling, formalized in appendix A).

‡ pair., exp. shows # of pairing and exponentiation computation (in $\mathbb{G}$ or $\mathbb{G}_T$), respectively.

7 Concluding Remarks

This paper presented the first results for expressive KP-ABE schemes with constant-size ciphertexts. In the future, it will be interesting to see if shorter private keys can be obtained without affecting the expressivity or the size of ciphertexts and to construct adaptively secure such schemes. Another challenging problem is to achieve similar results in the expressive ciphertext-policy setting.

References

1. M. Abdalla, E. Kiltz, G. Neven. Generalized Key Delegation for Hierarchical Identity-Based Encryption. In *ESORICS'07*, *LNCS* 4734, pp. 139–154. Springer.

2. N. Attrapadung, H. Imai. Dual-Policy Attribute Based Encryption. In *ACNS'09*, *LNCS* 5536, pp. 168–185, 2009.
3. N. Attrapadung, H. Imai. Conjunctive Broadcast and Attribute-Based Encryption. In *Pairing'09*, *LNCS* 5671, pp. 248–265, 2009.
4. N. Attrapadung, B. Libert. Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In *PKC'10*, *LNCS* 6056, pp. 384–402. Springer, 2010.
5. J. Bethencourt, A. Sahai, B. Waters. Ciphertext-Policy Attribute-Based Encryption. IEEE Symposium on Security and Privacy (S&P), pp. 321-334, 2007.
6. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238, 2004.
7. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical Identity-Based encryption with Constant Size Ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005.
8. D. Boneh, M. Franklin. Identity-Based Encryption from the Weil Pairing. In *SIAM* Journal of Computing 32(3), pp. 586–615, 2003. Earlier version in *Crypto'01*, *LNCS* 2139, pp. 213–229, 2001.
9. D. Boneh, C. Gentry, B. Waters. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *Crypto'05*, *LNCS* 3621, pp. 258–275, 2005.
10. D. Boneh, M. Hamburg. Generalized Identity Based and Broadcast Encryption Schemes. In *Asiacrypt'08*, *LNCS* 5350, pp. 455–470, 2008.
11. D. Boneh, A. Sahai, B. Waters. Functional Encryption: Definitions and Challenges. In *TCC'11*, to appear, 2011.
12. X. Boyen. General *Ad Hoc* Encryption from Exponent Inversion IBE. In *Eurocrypt'07*, *LNCS* 4515, pp. 394–411, 2007.
13. R. Canetti, S. Halevi, J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *Eurocrypt'03*, *LNCS* 2656, pp. 254–271, 2003.
14. R. Canetti, S. Halevi, J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Eurocrypt'04*, *LNCS* 3027, pp. 207–222, 2004.
15. M. Chase. Multi-authority Attribute Based Encryption. In *TCC'07*, *LNCS* 4392, pp. 515–534, 2007
16. M. Chase, S. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *ACM-CCS'09*, pp. 121–130, 2009.
17. J.-H. Cheon. Security Analysis of the Strong Diffie-Hellman Problem. In *Eurocrypt'06*, *LNCS* 4004, pp. 1–11, 2006.
18. L. Cheung, C. Newport. Provably secure ciphertext policy ABE. In *ACM-CCS'07*, pp. 456–465, 2007.
19. K. Emura, A. Miyaji, A. Nomura, K. Omote, M. Soshi. A Ciphertext-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length. In *ISPEC '09*, *LNCS* 5451, pp. 13–23, 2009.
20. V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS'06*, pp. 89–98, 2006.
21. V. Goyal, A. Jain, O. Pandey, A. Sahai. Bounded Ciphertext Policy Attribute Based Encryption. ICALP (2) 2008, *LNCS* 5126, pp. 579–591, 2008.
22. J. Herranz, F. Laguillaumie, C. Ràfols. Constant-Size Ciphertexts in Threshold Attribute-Based Encryption. In *PKC'10*, *LNCS* 6056, Springer, 2010.
23. J. Katz, A. Sahai, B. Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *Eurocrypt'08*, *LNCS* 4965, pp. 146-162.
24. A. Lewko, A. Sahai, B. Waters. Revocation Systems with Very Small Private Keys. In IEEE Symposium on Security and Privacy (S&P) 2010.
25. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Eurocrypt 2010*, *LNCS* series.
26. M. Naor. On Cryptographic Assumptions and Challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109, 2003.
27. M. Naor, B. Pinkas. Efficient Trace and Revoke Schemes. In *Financial Cryptography 2000*, *LNCS* 1962, pp. 1-20, 2000.
28. R. Ostrovsky, A. Sahai, B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM CCS'07*, pp. 195–203, 2007.
29. A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt'05*, *LNCS* 3494, pp. 457–473, 2005.
30. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Crypto'84*, *LNCS* 196, pp. 47–53, 1984.
31. T. Okamoto, K. Takashima, Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO'10*, *LNCS* 6223, pp. 191-208, 2010.
32. B. Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC 2011*.
33. S. Yamada, N. Attrapadung, G. Hanaoka, N. Kunihiro. Generic Constructions for Chosen-Ciphertext Secure Attribute Based Encryption. In *PKC 2011*.

A Variant: KP-ABE with Labeling

We re-formalize the KP-ABE definition of [31] in our context as follows. Intuitively, the difference from normal KP-ABE is that an attribute is required to be labeled with a number $j \in [1, n]$ and that each attribute in the set associated to a ciphertext is required to be labeled uniquely, namely 1 to n. The scheme of [31] further restricts the maximum repetition allowed for labels in one policy, which we denote by φ in Table 2.

Definition 8 (KP-ABE with labeling). *Let U be an attribute space and let a positive integer $n \in \mathbb{N}$. Define $U' = \{(j, u) \mid j \in [1, n], u \in U\}$. Define the ciphertext index domain as*

$$\Sigma_e^{\mathsf{KP}'} = \{\{(1, u_1), \ldots, (n, u_n)\} \mid u_1, \ldots, u_n \in U\}.$$

A KP-ABE with labeling for a collection $\mathcal{AS}'$ of access structures over U' is a functional encryption for $R^{\mathsf{KP}'} : \mathcal{AS}' \times \Sigma_e^{\mathsf{KP}'} \to \{0, 1\}$ defined by $R^{\mathsf{KP}'}(\mathbb{A}, \omega) = 1$ iff $\omega \in \mathbb{A}$ (for $\omega \in \Sigma_e^{\mathsf{KP}'}, \mathbb{A} \in \mathcal{AS}'$).

We observe that KP-ABE with large universe $U = \{0, 1\}^*$, *e.g.,* [20, 28] and ours, implies KP-ABE with labeling. This is since $U' \subset U$, $\Sigma_e^{\mathsf{KP}'} \subset \Sigma_e^{\mathsf{KP}}$, $\Sigma_k^{\mathsf{KP}'} \subset \Sigma_k^{\mathsf{KP}}$, and $R^{\mathsf{KP}'} \Leftrightarrow R^{\mathsf{KP}}$ holds and the implication comes from the embedding lemma [10, 4]. To the best of our knowledge, the converse is yet known to hold.

We now briefly propose a KP-ABE that conforms with the normal definition by modifying [31]. We construct by instantiating the general KP-FE scheme of [31] with $d = 1$, and with the inner product relation being instantiated to IBBE, similarly as we did in section 3.3, and setting the bound $\varphi = n$.

B Proof of Theorem 1

We construct a simple IND-sID-CPA adversary $\mathcal{B}$ against the IBBE scheme assuming that a selective-set attacker $\mathcal{A}$ has non-negligible advantage against the KP-ABE system. Namely, $\mathcal{B}$ plays the role of $\mathcal{A}$'s challenger and interacts with her own challenger in the IBBE security game.

The game begins with the KP-ABE adversary $\mathcal{A}$ choosing an attribute set $\omega^\star$ that she intends to attack. The IBBE adversary $\mathcal{B}$ then announces $S^\star = \{i \in \omega^\star\}$ as her target set of receivers. The system-wide IBBE public key that $\mathcal{B}$ receives from her challenger are relayed to $\mathcal{A}$ as system-wide parameters for the KP-ABE scheme.

Throughout the game, $\mathcal{A}$ may ask for the private key of any access structure (L, π) such that $\omega^\star$ does not satisfy (L, π). To answer such a query, $\mathcal{B}$ proceeds as follows. Let $L_{\omega^\star}$ be the sub-matrix formed by the rows of L that correspond to an attribute in $\omega^\star$. Since $\mathbf{1} = (1, 0, \ldots, 0)^\top$ is not in the row space of $L_{\omega^\star}$, there must exist an efficiently computable vector $\boldsymbol{w}$ such that $L_{\omega^\star} \cdot \boldsymbol{w} = \mathbf{0}$ and $\langle \mathbf{1}, \boldsymbol{w} \rangle \neq 0$ (according to proposition 1 in [20]). Let h denote the value $\langle \mathbf{1}, \boldsymbol{w} \rangle$. To construct a private key, $\mathcal{B}$ has to define a vector $\boldsymbol{u} = \alpha \cdot \boldsymbol{\beta}$ such that $\langle \mathbf{1}, \boldsymbol{u} \rangle = \alpha$. As in the proof of theorem 3 in [20], $\mathcal{B}$ implicitly sets $\boldsymbol{u}$ as $\boldsymbol{u} = \boldsymbol{v} + \psi \cdot \boldsymbol{w}$, where $\boldsymbol{v} = (v_1, \ldots, v_k)^\top$ is a randomly chosen vector and $\psi = (\alpha - v_1)/h$, so that $\langle \mathbf{1}, \boldsymbol{u} \rangle = \alpha$. To generate triples $(D_{i,1}, D_{i,2}, D_{i,3})$ for each row of L, $\mathcal{B}$ proceeds as follows.

1. Let $\Gamma_1 = \{j \in \{1, \ldots, \ell\} \mid \pi(j) \in \omega^\star\}$. For each $j \in \Gamma_1$, if $\boldsymbol{L}_j^\top = (m_{j1}, \ldots, m_{jk})$ denotes the j^{th} row of L, we have $\langle \boldsymbol{L}_j, \boldsymbol{u} \rangle = \langle \boldsymbol{L}_j, \boldsymbol{v} \rangle = \sum_{t_1=1}^{k} m_{jt_1} v_{t_1}$ and the share $\lambda_j = \langle \boldsymbol{L}_j, \boldsymbol{u} \rangle$ is thus computable, so that $\mathcal{B}$ can pick integers $\lambda_j, r_j \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and define

$$D_j = (D_{j,1}, D_{j,2}, D_{j,2}) = \big(g^{\lambda_j} \cdot f_1(\pi(j))^{r_j},\ g^{r_j},\ f_2(\pi(j))^{r_j}\big).$$

2. Let $\mathit{\Gamma}_2 = \{j \in \{1, \dots, \ell\} \mid \pi(j) \notin \omega^\star\}$. For each $j \in \mathit{\Gamma}_2$, $\mathcal{B}$ is allowed to query its own challenger to extract $(d_{j,1}, d_{j,2}, d_{j,3}) \leftarrow \mathit{\Pi}_{\mathsf{IBBE}}.\mathsf{Keygen}(\mathsf{msk}, \pi(j))$. Also, we have

$$\langle \boldsymbol{L}_j, \boldsymbol{u} \rangle = \langle \boldsymbol{L}_j, \boldsymbol{v} \rangle + \psi \cdot \langle \boldsymbol{L}_j, \boldsymbol{w} \rangle = \sum_{t_1=1}^{k} m_{jt_1} \Big(v_{t_1} + \frac{(\alpha - v_1)}{h} \cdot w_{t_1} \Big) = \mu_1 \cdot \alpha + \mu_2,$$

where the coefficients $\mu_1 = (\sum_{t_1=1}^{k} m_{jt_1} w_{t_1}) \cdot h^{-1}$ and $\mu_2 = h^{-1} \cdot \sum_{t_1=1}^{k} m_{jt_1} \big(h v_{t_1} - v_1 w_{t_1} \big)$ are both computable, so that $\mathcal{B}$ can obtain a well-formed triple $D_j = (D_{j,1}, D_{j,2}, D_{j,3})$ by setting

$$D_j = (D_{j,1}, D_{j,2}, D_{j,3}) = \Big(d_{j,1}^{\mu_1} \cdot g^{\mu_2} \cdot f_1(\pi(j))^{r'_j}, d_{j,2}^{\mu_1} \cdot g^{r'_j}, d_{j,3}^{\mu_1} \cdot f_2(\pi(j))^{r'_j} \Big).$$

When $\mathcal{A}$ decides to enter the challenge phase, she outputs messages $\mathsf{M}_0, \mathsf{M}_1$ that $\mathcal{B}$ forwards to her challenger before relaying the challenge ciphertexts back to $\mathcal{A}$.

The second series of private key queries is handled as the first one and $\mathcal{B}$ eventually outputs the same result $d' \in \{0, 1\}$ as $\mathcal{A}$ does. It is easy to see that $\mathcal{B}$ never has to query her challenger to extract the private key for an identity of the target attribute set $S^\star = \omega^\star$. It comes that $\mathcal{B}$ is successful whenever $\mathcal{A}$ is so. □

C The Boneh-Hamburg Spatial Encryption and IBBE Schemes

We recall the concept of spatial encryption [10]. For a matrix $M \in \mathbb{Z}_p^{n \times d}$ and a vector $\boldsymbol{c} \in \mathbb{Z}_p^n$, one considers the affine space $\mathrm{Aff}(M, \boldsymbol{c}) = \{M\boldsymbol{w} + \boldsymbol{c} \mid \boldsymbol{w} \in \mathbb{Z}_p^d\}$. Let $\mathcal{V}_n \subseteq 2^{(\mathbb{Z}_p^n)}$ be the collection of all affine spaces inside $\mathbb{Z}_p^n$. That is, $\mathcal{V}_n$ is defined as

$$\mathcal{V}_n = \{\mathrm{Aff}(M, \boldsymbol{c}) \mid M \in \mathbb{M}_{n \times d}, c \in \mathbb{Z}_p^n, d \leq n\},$$

where $\mathbb{M}_{n \times d}$ is the set of all $n \times d$ matrices in $\mathbb{Z}_p$.

In a spatial encryption scheme, private keys correspond to affine subspaces and ciphertexts are associated with a vector and can be decrypted by any private key associated with a subspace containing that vector. In addition, a private key corresponding to an affine subspace V_1 allows deriving (using algorithm $\mathsf{Delegate}$ below) a private key for any subspace V_2 such that $V_2 \subset V_1$.

In [10], Boneh and Hamburg gave a construction of spatial encryption with short ciphertexts. It is inspired by the Boneh-Boyen-Goh hierarchical identity-based encryption scheme [7].

$\mathsf{Setup}(\lambda, n)$: given a security parameter $\lambda \in \mathbb{N}$ and a maximal dimension $n \in \mathbb{N}$ for affine subspaces, choose prime-order bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ and $g \stackrel{R}{\leftarrow} \mathbb{G}$. Choose $\alpha, \alpha_0 \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^\top \stackrel{R}{\leftarrow} \mathbb{Z}_p^n$ to compute $h_0 = g^{\alpha_0}$, $\boldsymbol{H} = g^{\boldsymbol{\alpha}}$ and $e(g, g)^\alpha$. The master public key is $\mathsf{mpk} = (g,\ e(g, g)^\alpha,\ h_0,\ \boldsymbol{H} = g^{\boldsymbol{\alpha}})$ while the master secret key is $\mathsf{msk} = (\boldsymbol{\alpha}, \alpha_0, \alpha)$.

$\mathsf{Keygen}(\mathsf{msk}, V)$: to generate a key for an affine space $V = \mathrm{Aff}(M, \boldsymbol{x})$, choose $r \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and compute

$$K_V = (K_1, K_2, K_3) = \big(g^\alpha \cdot h_0^r \cdot g^{r \langle \boldsymbol{x}, \boldsymbol{\alpha} \rangle},\ g^r,\ g^{r M^\top \boldsymbol{\alpha}} \big)$$

$\mathsf{Delegate}(\mathsf{msk}, V_1, K_{V_1}, V_2)$: takes as input two subspaces $V_1 = \mathrm{Aff}(M_1, \boldsymbol{x_1})$, $V_2 = \mathrm{Aff}(M_2, \boldsymbol{x_2})$. It outputs $\perp$ if $V_2 \not\subset V_1$. Otherwise, we must have $M_2 = M_1 T$ and $\boldsymbol{x_2} = \boldsymbol{x_1} + M_1 \boldsymbol{y}$ for some efficiently computable matrix T and vector $\boldsymbol{y}$. Given $K_{V_1} = (K_1, K_2, K_3)$, these allow computing

$$\begin{aligned} K_{V_2} &= \big(K_1 \cdot K_3^{\boldsymbol{y}^\top} \cdot h_0^{r_1} \cdot g^{r_1 \langle \boldsymbol{x_2}, \boldsymbol{\alpha} \rangle},\ K_2 \cdot g^{r_1},\ K_3^{T^\top} \cdot g^{r_1 M_2^\top \boldsymbol{\alpha}} \big) \\ &= \big(g^\alpha \cdot h_0^{r'} \cdot g^{r' \langle \boldsymbol{x_2}, \boldsymbol{\alpha} \rangle},\ g^{r'},\ g^{r' M_2^\top \boldsymbol{\alpha}} \big), \end{aligned}$$

where $r' = r + r_1$, for some randomly drawn $r_1 \stackrel{R}{\leftarrow} \mathbb{Z}_p$.

$\mathsf{Encrypt}(\mathsf{mpk}, \boldsymbol{x}, \mathsf{M})$: to encrypt $\mathsf{M} \in \mathbb{G}_T$ for the vector $\boldsymbol{x} \in Z_p^n$, choose $s \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute

$$C = (C_0, C_1, C_2) = \big(m \cdot e(g,g)^{\alpha s},\ g^s,\ h_0^s \cdot g^{s\langle \boldsymbol{x}, \boldsymbol{\alpha} \rangle}\big)$$

$\mathsf{Decrypt}(\mathsf{mpk}, V, K_V, \boldsymbol{x}, C)$: parse C as (C_0, C_1, C_2) and K_V as (K_1, K_2, K_3). Compute the plaintext as $\mathsf{M} = C_0 \cdot e(C_2, K_2)/e(C_1, K_1)$.

The Boneh-Hamburg IBBE is a particular case of the spatial encryption primitive and its specification is as follows.

$\mathsf{Setup}(\lambda, n)$: given a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathbb{N}$ such that $n-1$ is the maximal number of receivers per ciphertext, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ and $g \overset{R}{\leftarrow} \mathbb{G}$. Choose $h_0 \overset{R}{\leftarrow} \mathbb{G}$ as well a vector $\boldsymbol{h} = (h_1, \ldots, h_n)^\top \overset{R}{\leftarrow} \mathbb{G}^n$ such that $h_i = g^{a_i}$ for $i = 1, \ldots, n$ with $\boldsymbol{a} = (a_1, \ldots, a_n)^\top \overset{R}{\leftarrow} \mathbb{Z}_p^n$. Finally, pick $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and compute $e(g,g)^\alpha$. The master public key is $\mathsf{mpk} = \big(g,\ e(g,g)^\alpha,\ h_0,\ \boldsymbol{h} = g^{\boldsymbol{a}}\big)$. while the master secret key is $\mathsf{msk} = (\boldsymbol{a}, \alpha)$.

$\mathsf{Keygen}(\mathsf{msk}, \mathsf{ID})$: to generate a private key for an identity ID, choose $r \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and compute a tuple

$$K_{\mathsf{ID}} = (K_1, K_2, T_1, \ldots, T_{n-1}) = \big(g^\alpha \cdot h_0^r,\ g^r,\ h_2^r \cdot h_1^{-\mathsf{ID} \cdot r}, \ldots,\ h_n^r \cdot h_{n-1}^{-\mathsf{ID} \cdot r}\big)$$

for which $(T_1, \ldots, T_{n-1})^\top$ can be written $g^{r \cdot M_{\mathsf{ID}}^\top \cdot \boldsymbol{a}}$, for some matrix $M_{\mathsf{ID}} \in \mathbb{Z}_p^{n \times (n-1)}$ (defined below).

$\mathsf{Encrypt}(\mathsf{mpk}, S, \mathsf{M})$: to encrypt $\mathsf{M} \in \mathbb{G}_T$ for the receiver set $S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_q\}$, where $q \leq n-1$,

1. Expand the polynomial $P[Z] = \prod_{\mathsf{ID}_i \in S}(Z - \mathsf{ID}_i)$ into $P[Z] = \rho_{q+1} Z^q + \rho_q Z^{q-1} + \cdots + \rho_2 Z + \rho_1$.
2. Pick $s \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and compute

$$C = (C_0, C_1, C_2) = \Big(\mathsf{M} \cdot e(g,g)^{\alpha \cdot s},\ g^s,\ \big(h_0 \cdot h_1^{\rho_1} \cdots h_{q+1}^{\rho_{q+1}}\big)^s\Big).$$

$\mathsf{Decrypt}(\mathsf{msk}, K_{\mathsf{ID}}, C)$: parse S as $\{\mathsf{ID}_1, \ldots, \mathsf{ID}_q\}$, the ciphertext C as (C_0, C_1, C_2) and K_{ID} as

$$K_{\mathsf{ID}} = (K_1, K_2, T_1, \ldots, T_{n-1}) \in \mathbb{G}^{n+1}.$$

1. Let $i \in \{1, \ldots, q\}$ be the position of ID in S. Expand $P_i[Z] = \prod_{\mathsf{ID}_j \in S \setminus \{\mathsf{ID}_i\}}(Z - \mathsf{ID}_j)$ into

$$P_i[Z] = y_q^{(i)} Z^{q-1} + y_{q-1}^{(i)} Z^{q-2} + \cdots + y_2^{(i)} Z + y_1^{(i)}$$

and compute $(D_{\mathsf{ID}}, d_{\mathsf{ID}}) \in \mathbb{G}^2$ as

$$(D_{\mathsf{ID}}, d_{\mathsf{ID}}) = \big(K_1 \cdot T_1^{y_1^{(i)}} \cdot T_2^{y_2^{(i)}} \cdots T_q^{y_q^{(i)}},\ K_2\big) = \Big(g_2^\alpha \cdot \big(h_0 \cdot h_1^{\rho_1} \cdots h_{q+1}^{\rho_{q+1}}\big)^r,\ g^r\Big)$$

where $\rho_1, \ldots, \rho_{q+1}$ are the coefficients of the polynomial $P[Z]$ (calculated as per step 1 of the encryption algorithm).

2. Recover $\mathsf{M} = C_0 \cdot e\big(C_1, D_{\mathsf{ID}}\big)^{-1} \cdot e\big(C_2,\ d_{\mathsf{ID}}\big)$.

To explain the first step of the decryption algorithm, one observes that, for any two polynomials $(Z-\mathsf{ID})$ and $P_i[Z] = y_q^{(i)}Z^{q-1} + y_{q-1}^{(i)}Z^{q-2} + \cdots + y_2^{(i)}Z + y_1^{(i)}$, the coefficients of their product $P[Z] = (Z-\mathsf{ID})P_i[Z] = \sum_{i=1}^{q+1} \rho_i Z^{i-1}$ are given by

$$\boldsymbol{\rho} = \begin{pmatrix}\rho_1 \; \rho_2 \cdots \; \rho_{q+1}\end{pmatrix}^\top = M_{\mathsf{ID}} \cdot \boldsymbol{y} = \begin{pmatrix} -\mathsf{ID} & & & & \\ 1 & -\mathsf{ID} & & & \\ & 1 & -\mathsf{ID} & & \\ & & \ddots & \ddots & \\ & & & 1 & -\mathsf{ID} \\ & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1^{(i)} \\ y_2^{(i)} \\ \vdots \\ y_q^{(i)} \end{pmatrix},$$

where $M_{\mathsf{ID}} \in \mathbb{Z}_p^{(q+1)\times q}$. Since the latter matrix is such that

$$M_{\mathsf{ID}}^\top \cdot \boldsymbol{a}|_{q+1} = M_{\mathsf{ID}}^\top \cdot \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{q+1} \end{pmatrix} = \begin{pmatrix} a_2 - \mathsf{ID} \cdot a_1 \\ a_3 - \mathsf{ID} \cdot a_2 \\ \vdots \\ a_{q+1} - \mathsf{ID} \cdot a_q \end{pmatrix},$$

for each private key K_{ID}, the delegation components satisfy

$$(T_1, \ldots, T_q)^\top = \left(h_2^r \cdot h_1^{-\mathsf{ID}\cdot r},\; h_3^r \cdot h_2^{-\mathsf{ID}\cdot r}, \ldots,\; h_{q+1}^r \cdot h_q^{-\mathsf{ID}\cdot r}\right)^\top = g^{rM_1^\top \cdot \boldsymbol{a}}.$$

Therefore, since $\rho = M_{\mathsf{ID}} \cdot \boldsymbol{y}$, we have

$$(h_0 \cdot \prod_{k=1}^{q+1} h_k^{\rho_k})^r = h_0^r \cdot g^{r\cdot\boldsymbol{\rho}^\top \cdot \boldsymbol{a}|_{q+1}} = h_0^r \cdot g^{r\boldsymbol{y}^\top \cdot M_{\mathsf{ID}}^\top \cdot \boldsymbol{a}|_{q+1}} = h_0^r \cdot T_1^{y_1^{(i)}} \cdots T_q^{y_q^{(i)}}$$

which explains why $(D_{\mathsf{ID}}, d_{\mathsf{ID}})$ are correctly calculated at step 1 of the decryption algorithm. To explain step 2 of the decryption algorithm, we note that, for each $\mathsf{ID} \in S$, the pair $(D_{\mathsf{ID}}, d_{\mathsf{ID}})$ satisfies

$$e(D_{\mathsf{ID}}, g) = e(g,g)^\alpha \cdot e(h_0 \cdot h_1^{\rho_1} \cdots h_{q+1}^{\rho_{q+1}}, d_{\mathsf{ID}}) \tag{5}$$

By raising both members of (5) to the power $s \in \mathbb{Z}_p^*$, where s is the random encryption exponent, we see why M can be recovered at decryption.

The security of this scheme was proved [10] in the selective-ID model under the n-DBDHE assumption. The construction is easily seen to fit the general IBBE template.

The security of the (somewhat simpler) IBBE scheme of section 3.3 under the n-DBDHE assumption follows from the fact that the underlying inner product encryption scheme can be casted as an instance of the above spatial encryption system. Indeed, as shown in [4], a vector $\boldsymbol{X} = (x_1, \ldots, x_n)^\top$ of key attributes can be mapped onto a $(n-1)$-dimension affine space $V_{\boldsymbol{X}} = \mathrm{Aff}(M_{\boldsymbol{X}}, \boldsymbol{0}_n) = \{M_{\boldsymbol{X}}\boldsymbol{w} + \boldsymbol{0}_n \mid \boldsymbol{w} \in \mathbb{Z}_p^{n-1}\}$ with the matrix $M_{\boldsymbol{X}} \in \mathbb{Z}_p^{n\times(n-1)}$

$$M_{\boldsymbol{X}} = \begin{pmatrix} -\frac{x_2}{x_1} & -\frac{x_3}{x_1} & \cdots & -\frac{x_n}{x_1} \\ & I_{n-1} & & \end{pmatrix}.$$

From there, it is easy to see that, for any vector $\boldsymbol{Y} = (y_1, \ldots, y_n)^\top$, we have the equivalence $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = 0 \;\Leftrightarrow\; \boldsymbol{Y} \in V_{\boldsymbol{X}}$, which is immediate from

$$\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = 0 \Leftrightarrow y_1 = y_2 \cdot (-\frac{x_2}{x_1}) + \cdots + y_n \cdot (-\frac{x_n}{x_1}) \tag{6}$$

$$\Leftrightarrow \boldsymbol{Y} = M_{\boldsymbol{X}} \cdot (y_2, \ldots, y_n)^\top \;\Leftrightarrow\; \boldsymbol{Y} \in V_{\boldsymbol{X}}. \tag{7}$$

D Proof of Theorem 3

We outline an algorithm $\mathcal{B}$ that receives as input $(g, h, z_1, \dots, z_n, z_{n+2}, \dots, z_{2n}, T) \in \mathbb{G}^{2n+1} \times \mathbb{G}_T$, where $z_i = g^{(\gamma^i)}$, and decides if $T = e(g,h)^{(\gamma^{n+1})}$ using the selective-set adversary $\mathcal{A}$. We define $\boldsymbol{\gamma} = (\gamma, \gamma^2, \dots, \gamma^n)^\top$ for further use.

At the very beginning of the attack game, $\mathcal{A}$ announces the attribute set $\omega^\star$ that she wishes to be challenged upon. This set $\omega^\star$ is used to define a vector $\boldsymbol{Y} = (y_1, \dots, y_n)^\top$ as the coefficients of $P_{\omega^\star}[Z] = \prod_{j \in \omega^\star}(Z - j) = \sum_{i=1}^{n} y_i Z^{i-1}$ (in the event that $|\omega^\star| = q$ is strictly smaller than $n-1$, $\mathcal{B}$ sets $y_{q+1} = \dots = y_n = 0$).

To simulate the master public key, it will generate according to three parts: the first part relates to non-negated attributes, which are elements $\boldsymbol{U} = g^{\boldsymbol{\beta}}$; the second part relates to negated attributes, which are elements $\boldsymbol{H} = g^{\boldsymbol{\alpha}}$; the last part is the common element $e(g,g)^\alpha$.

- For the common part, it picks $\delta_0 \overset{R}{\leftarrow} \mathbb{Z}_p$ and lets $e(g,g)^\alpha = e(z_1, z_n)^{\delta_0}$. This implicitly defines the master secret as $\alpha = \gamma^{(n+1)} \cdot \delta_0$.
- For the public key part related to non-negated attributes, it simulates similarly as in the proof of the underlying IBBE of section 3.3 (which we omitted the proof there). More concretely, it picks $\theta_0 \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $u_0 = g^{\theta_0} \cdot g^{-\langle \boldsymbol{\gamma}, \boldsymbol{Y} \rangle}$ from $g^{\boldsymbol{\gamma}}$. Other components of $\boldsymbol{U}$ are defined by setting $\boldsymbol{U'} := (u_1, \dots, u_n)^\top = g^{\boldsymbol{\gamma}} \cdot g^{\boldsymbol{\theta}}$, for some randomly chosen vector $\boldsymbol{\theta} \overset{R}{\leftarrow} \mathbb{Z}_p^n$, so that we have $\boldsymbol{\beta'} := (\beta_1, \dots, \beta_n)^\top = \boldsymbol{\gamma} + \boldsymbol{\theta}$.
- For the public key part related to negated attributes, it simulates similarly as in the proof of the underlying IBR of section 4 (which is recorded in the proof of theorem 2). Intutively, it proceeds as if the announced set $\tilde{S}$ in theorem 2 (for private key queries there) is set to $\tilde{S} = \omega^\star$. More concretely, we first write $\omega^\star = \{\omega_1, \dots, \omega_q\}$ in some order, then we define their corresponding vectors $\boldsymbol{X}_1, \dots, \boldsymbol{X}_q$ as $\boldsymbol{X}_k = (1, \omega_k, \dots, \omega_k^{n-1})^\top$. It then defines the $n \times n$ matrix $B = \big(\boldsymbol{b}_1 | \dots | \boldsymbol{b}_q | \boldsymbol{0} | \dots | \boldsymbol{0}\big)$ from the definition of $\boldsymbol{b}_k$ as in equation (4), where it can be re-written this time as:

$$\boldsymbol{b}_k^\top \cdot M_{\boldsymbol{X_k}} = \boldsymbol{b}_k^\top \cdot \begin{pmatrix} -\omega_k & -\omega_k^2 & \cdots & -\omega_k^{n-1} \\ & & I_{n-1} & \end{pmatrix} = \boldsymbol{0}.$$

It then proceeds to define $\boldsymbol{H}$ as $\boldsymbol{H} = g^{B \cdot \boldsymbol{a}} \cdot g^{\boldsymbol{\delta}}$, for known random $\boldsymbol{\delta} \overset{R}{\leftarrow} \mathbb{Z}_p^n$. We also recall that $\boldsymbol{a} = (\gamma^n, \gamma^{n-1}, \dots, \gamma)^\top$.

At any time, the adversary $\mathcal{A}$ may query a private key for arbitrary access structures $\tilde{\mathbb{A}}$ such that $R^{\mathsf{KP}}(\tilde{\mathbb{A}}, \omega^\star) = 0$. By assumption, $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$, defined over a set $\mathcal{P}$ of parties, associated with a linear secret sharing scheme $\mathit{\Pi}$. Let $L \in \mathbb{Z}_p^{\ell \times n}$ denote the share-generating matrix for $\mathit{\Pi}$. Since $R^{\mathsf{KP}}(\tilde{\mathbb{A}}, \omega^\star) = 0$, we have that $R^{\mathsf{KP}}(\mathbb{A}, \omega') = 0$, where $\omega' = N(\omega^\star)$. Therefore, $\boldsymbol{1} = (1, 0, \dots, 0)^\top$ does not lie in the row space of $L_{\omega'}$, which is the sub-matrix of L formed by rows corresponding to attributes in ω'. Hence, similarly to the proof of Theorem 1, due to the proposition 11 in [24], we have that there must exist an efficiently computable vector $\boldsymbol{w} \in \mathbb{Z}_p^n$ such that $\langle \boldsymbol{1}, \boldsymbol{w} \rangle = 1$ and $L_{\omega'} \cdot \boldsymbol{w} = \boldsymbol{0}$. Now $\mathcal{B}$ will implicitly define each share of α as $\lambda_i = \langle \boldsymbol{L_i}, \boldsymbol{v} \rangle$, corresponding to a party named $\breve{x}_i \in \mathcal{P}$ where x_i is the underlying attribute ($\breve{x}_i$ being primed or unprimed). It does by implicitly defining $\boldsymbol{v} = \boldsymbol{\zeta} + (\alpha - \zeta_1)\boldsymbol{w}$ where $\boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_n)^\top \overset{R}{\leftarrow} \mathbb{Z}_p^n$. Note that we have that $v_1 = \alpha$ and that $v_2, \dots, v_n \in \mathbb{Z}_p$ are uniformly distributed, as required in Definition 3. Although $\mathcal{B}$ cannot compute $\langle \boldsymbol{L_i}, \boldsymbol{v} \rangle$ for all i, it can compute a private key as follows.

- For negated parties $\breve{x}_i = x'_i$, $\mathcal{B}$ distinguishes two cases.
  - If $x_i \in \omega^\star$ (and thus $\breve{x}_i \notin \omega'$), $\lambda_i = \langle \boldsymbol{L_i}, \boldsymbol{v} \rangle$ depends on α and can be written as $\lambda_i = \nu_1 \alpha + \nu_2$ for constants $\nu_1, \nu_2 \in \mathbb{Z}_p$ that are known to $\mathcal{B}$. Since in this case $x_i \in \omega^\star = \{\omega_1, \ldots, \omega_q\}$, hence $x_i = \omega_k$ for some $k \in [1, q]$. Now recall that the underlying IBR scheme allows us to simulate the IBR key for identity $\omega_1, \ldots, \omega_q$. Hence, the one for ω_k can also be constructed and is of the form

$$(D_1, D_2, K_2, \ldots, K_n) = \Big(g^\alpha \cdot h_1^r,\ g^r,\ \big(h_1^{-\frac{\rho_{i,2}}{\rho_{i,1}}} \cdot h_2\big)^r, \ldots,\ \big(h_1^{-\frac{\rho_{i,n}}{\rho_{i,1}}} \cdot h_n\big)^r \Big),$$

    where $\boldsymbol{\rho_i} = (\rho_{i,1}, \ldots, \rho_{i,n})^\top = \boldsymbol{X}_k = (1, \omega_k, \ldots, \omega_k^{n-1}) = (1, x_i, \ldots, x_i^{n-1})$, for some (unknown) randomness $r \in \mathbb{Z}_p$.

    From there, $\mathcal{B}$ can obtain a valid piece of key material $(D_{i,1}^{(2)}, D_{i,2}^{(2)}, K_{i,2}^{(2)}, \ldots, K_{i,n}^{(2)})$ by drawing $r' \xleftarrow{R} \mathbb{Z}_p$ and setting $D_{i,1}^{(2)} = D_1^{\nu_1} \cdot g^{\nu_2} \cdot h_1^{r'}$, $D_{i,2}^{(2)} = D_2^{\nu_1} \cdot g^{r'}$ and $K_{i,j}^{(2)} = K_j^{\nu_1} \cdot (h_1^{-\rho_{i,j}/\rho_{i,1}} \cdot h_j)^{r'}$ for each $j \in \{2, \ldots, n\}$.
  - If $x_i \notin \omega^\star$ (so that $\breve{x}_i \in \omega'$), $\langle \boldsymbol{L_i}, \boldsymbol{w} \rangle = \boldsymbol{0}$ so that $\boldsymbol{L_i} \cdot \boldsymbol{v} = \boldsymbol{L_i} \cdot \boldsymbol{\zeta}$ is entirely known to $\mathcal{B}$ that can easily compute a suitably distributed tuple

$$D_i = (D_{i,1}^{(2)}, D_{i,2}^{(2)}, K_{i,2}^{(2)}, \ldots, K_{i,n}^{(2)}),$$

    where $D_{i,1}^{(2)} = g^{\boldsymbol{L_i} \cdot \boldsymbol{v}} \cdot h_1^{r_i}$ for a random $r_i \xleftarrow{R} \mathbb{Z}_p$.
- For non-negated parties $\breve{x}_i = x_i$, $\mathcal{B}$ proceeds as follows.
  - If $x_i \in \omega^\star$, $\lambda_i = \langle \boldsymbol{L_i}, \boldsymbol{v} \rangle$ does not depend on α and is entirely known to $\mathcal{B}$. Therefore, $\mathcal{B}$ can compute the key material

$$D_i = (D_{i,1}^{(1)}, D_{i,2}^{(1)}, K_{i,2}^{(1)}, \ldots, K_{i,n}^{(1)})$$

    by setting $D_{i,1}^{(1)} = g^{\lambda_i} \cdot u_0^{r_i}$ for random $r_i \xleftarrow{R} \mathbb{Z}_p$.
  - If $x_i \notin \omega^\star$, $\lambda_i = \langle \boldsymbol{L_i}, \boldsymbol{v} \rangle$ is of the form $\lambda_i = \nu_1 \alpha + \nu_2$ for known constants $\nu_1, \nu_2 \in \mathbb{Z}_p$ and $\mathcal{B}$ has to proceed as in [10][Theorem 1]. Namely, it considers the $n \times (n-1)$ matrix

$$M_{\boldsymbol{\rho_i}} = \begin{pmatrix} -\frac{\rho_{i,2}}{\rho_{i,1}} & -\frac{\rho_{i,3}}{\rho_{i,1}} & \cdots & -\frac{\rho_{i,n}}{\rho_{i,1}} \\ & I_{n-1} & & \end{pmatrix} = \begin{pmatrix} -x_i & -x_i^2 & \cdots & -x_i^{n-1} \\ & I_{n-1} & & \end{pmatrix}$$

    where $\rho_{i,j} = x_i^{j-1}$ for $j = 1$ to n. Since $x_i \notin \omega^\star$, the vector

$$\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^\top = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top$$

    is such that $\boldsymbol{\xi}^\top M_{\boldsymbol{\rho_i}} = \boldsymbol{0}$ but $\langle -\boldsymbol{Y}, \boldsymbol{\xi} \rangle \neq 0$. Using this fact, the simulator $\mathcal{B}$ can first generate a tuple of the form

$$(D_1, D_2, K_2, \ldots, K_n) = \Big(g^\alpha \cdot u_0^{\tilde{r}},\ g^{\tilde{r}},\ g^{\tilde{r} M_{\boldsymbol{\rho_i}}^\top \boldsymbol{\beta'}} \Big),$$

    with $\boldsymbol{\beta'} = (\beta_1, \ldots, \beta_n)^\top$ and where $\tilde{r}$ is defined as

$$\tilde{r} = r + \delta_0 (\xi_1 \gamma^n + \xi_2 \gamma^{n-1} + \cdots + \xi_n \gamma) / \langle \boldsymbol{Y}, \boldsymbol{\xi} \rangle.$$

To see why $\mathcal{B}$ is able to compute this, we note that, for any vector $\boldsymbol{f} \in \mathbb{Z}_p^n$ the coefficient of γ^{n+1} in the product $\tilde{r}\langle \boldsymbol{f}, \boldsymbol{\gamma} \rangle$ is $\delta_0 \langle \boldsymbol{f}, \boldsymbol{\xi} \rangle / \langle \boldsymbol{Y}, \boldsymbol{\xi} \rangle$. Given that $M_{\boldsymbol{\rho}_i}^{\top} \boldsymbol{\xi} = \boldsymbol{0}$, when $\boldsymbol{f}^{\top}$ is successively set as each row of $M_{\boldsymbol{\rho}_i}^{\top}$, the above argument shows that the unknown element $z_{n+1} = g^{(\gamma^{n+1})}$ is canceled out in $g^{\tilde{r} M_{\boldsymbol{\rho}_i}^{\top} \boldsymbol{\beta}'}$, which is thus computable from available elements. In addition, by applying the same argument to $\boldsymbol{f} = \boldsymbol{Y}$, we see that

$$g^{\alpha} \cdot u_0^{\tilde{r}} = z_{n+1}^{\delta_0} \cdot \left(g^{\theta_0} \cdot g^{-\langle \boldsymbol{\gamma}, \boldsymbol{Y} \rangle} \right)^{\tilde{r}}$$

is also computable since the coefficient of γ^{n+1} is $-\delta_0$ in the product $-\tilde{r}\langle \boldsymbol{\gamma}, \boldsymbol{Y} \rangle$. Once algorithm $\mathcal{B}$ has obtained $(D_1, D_2, K_2, \ldots, K_n)$, it easily obtains a suitably distributed tuple $(D_{i,1}^{(1)}, D_{i,2}^{(1)}, K_{i,2}^{(1)}, \ldots, K_{i,n}^{(1)})$ in the same way as for negated parties.

To generate the challenge ciphertext, $\mathcal{B}$ proceeds almost exactly as in the proof of theorem 2. Due to the choice of $\boldsymbol{U}$ and $\boldsymbol{H}$ in the setup phase, we have $u_0 \cdot g^{\langle \boldsymbol{\beta}', \boldsymbol{Y} \rangle} = g^{\theta_0 + \langle \boldsymbol{\theta}, \boldsymbol{Y} \rangle}$ and $g^{\langle \boldsymbol{\alpha}, \boldsymbol{Y} \rangle} = g^{\langle \boldsymbol{\delta}, \boldsymbol{Y} \rangle}$, so that the simulator $\mathcal{B}$ can flip a random coin $d \overset{R}{\leftarrow} \{0, 1\}$ and calculate

$$C_0 = \mathsf{M}_d \cdot T^{\delta_0}, \qquad C_1 = h, \qquad C_2 = h^{\theta_0 + \langle \boldsymbol{\theta}, \boldsymbol{Y} \rangle}, \qquad C_3 = h^{\langle \boldsymbol{\delta}, \boldsymbol{Y} \rangle}.$$

If $T = e(g, h)^{(\gamma^{n+1})}$, the ciphertext (C_0, C_1, C_2, C_3) is easily seen to form a valid encryption of M_d whereas it perfectly hides the bit $d \in \{0, 1\}$ if $T \in_R \mathbb{G}_T$. □