Attribute-Based Encryption Schemes with Constant-Size Ciphertexts

Nuttapong Attrapadung, Javier Herranz, Fabien Laguillaumie, Benoît Libert, Elie de Panafieu, Carla Ràfols

Attribute-Based Encryption Schemes with Constant-Size Ciphertexts ⋆

Nuttapong Attrapadung[1], Javier Herranz[2], Fabien Laguillaume[3], Benoît Libert[4], Elie de Panafieu[5], and Carla Ràfols[2]

[1] Research Center for Information Security, AIST (Japan)
[2] Universitat Politècnica de Catalunya, Dept. Matemàtica Aplicada IV (Spain)
[3] Université de Caen Basse-Normandie (France)
[4] Université catholique de Louvain, ICTEAM Institute – Crypto Group (Belgium)
[5] 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 dual ciphertext-policy scenario proceeds the other way around), 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 exception only supports restricted forms of access policies. This paper proposes the first attribute-based encryption (ABE) schemes allowing for truly expressive access structures and with constant ciphertext size. Our first result is a ciphertext-policy attribute-based encryption (CP-ABE) scheme with $O(1)$-size ciphertexts for threshold access policies and where private keys remain as short as in previous systems. As a second result, we show that a certain class of identity-based broadcast encryption schemes generically yields monotonic key-policy attribute-based encryption (KP-ABE) systems in the selective set model. Our final contribution is a KP-ABE realization supporting non-monotonic access structures (*i.e.*, that may contain negated attributes) with short ciphertexts. As an intermediate step towards this result, we describe a new efficient identity-based revocation mechanism that, when combined with a particular instantiation of our general monotonic construction, gives rise to the most expressive KP-ABE realization with constant-size ciphertexts. The downside of our second and third 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. Public-key cryptography, provable security, attribute-based encryption, access control, 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 [38] introduced attribute-based encryption (ABE), which refines identity-based encryption [40, 11] 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 [28] 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.

⋆ This paper merges and extends the results of two papers [30, 6] published in PKC 2010 and PKC 2011.

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 [7, 23]. The construction of Cheung and Newport [23] only handles AND gates while the first expressive construction [7] was only analyzed in the generic group model. Goyal, Jain, Pandey and Sahai [29] 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 [42] and one of them was recently extended by Lewko *et al.* [33], and subsequently by Okamoto and Takashima [41], into schemes providing adaptive security whereas all prior works on ABE were limited to deal with selective adversaries [17, 18, 8] – 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 [36] 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 [32] 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 [28, 36, 42, 33, 32] 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 [27] but, as in the Cheung-Newport realization [23], policies are restricted to a single AND gate.

This paper aims at devising ABE schemes with constant-size ciphertexts[6] (regardless of the number of underlying attributes) allowing for as expressive policies as possible. To this end, we propose several tradeoffs in terms of efficiency and expressivity.

Our first result is to design a CP-ABE system for threshold policies (namely, decryption works if the ciphertext and the receiver's private key have at least t attributes in common, where the threshold t is specified by the sender) with constant-size ciphertexts and where the private key size is linear in the number of attributes held by the user. The scheme belongs to the ciphertext-policy family in that the sender has the flexibility of choosing the threshold as he likes. The security is proved against selective adversaries under a non-interactive assumption.

As a second contribution, we 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 [17, 8]) 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.

[6] As in the literature on broadcast encryption (see, e.g., [12]) 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.

In a third 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 [36, 32]. 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 KP-ABE scheme, we design a new identity-based revocation (IBR) mechanism (as defined by Lewko, Sahai and Waters [32]) 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 [5] 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 [5] 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 ABE schemes with compact ciphertexts that can be proven secure against adaptive adversaries (as in the work of Lewko *et al.* [33]).

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 [20, 21] for a conjunctive setting and in [4] for a disjunctive setting. Besides the two usual flavors of ABE, another recently considered kind of ABE schemes [3], 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. Our CP-ABE scheme for threshold policies is described in Section 3. We describe our general construction of monotonic KP-ABE in Section 4. The new revocation scheme is depicted in Section 5.1, whereas the new non-monotonic ABE realization with compact ciphertexts is presented in Section 5.2. We compare the efficiency of some non-monotonic KP-ABE schemes in Section 5.3. Some concluding remarks and open problems are given in Section 6.

2 Background and Definitions

Notation. We will treat a vector as a column vector, unless stated otherwise. Namely, for any vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{Z}_p^n$, $g^{\vec{\alpha}}$ stands for the vector of group elements $(g^{\alpha_1}, \ldots, g^{\alpha_n})^\top \in \mathbb{G}^n$. For $\vec{a}, \vec{z} \in \mathbb{Z}_p^n$, we denote their inner product as $\langle \vec{a}, \vec{z} \rangle = \vec{a}^\top \vec{z} = \sum_{i=1}^n a_i z_i$. Given $g^{\vec{a}}$ and $\vec{z}$, $(g^{\vec{a}})^{\vec{z}} := g^{\langle \vec{a}, \vec{z} \rangle}$ is computable without knowing $\vec{a}$. 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 Secret sharing schemes

In this section we recall the definitions of access structures and linear secret sharing schemes, as defined in [28].

Definition 1 (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 ω, ω', if $\omega \in \mathbb{A}$ and $\omega' \subseteq \omega$, then $\omega \in \mathbb{A}$. 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.*

Threshold access structures are a specific case which appears very often in the cryptographic literature.

Definition 2 (Threshold Access Structure). *An access structure* $\mathbb{A}$ *is said to be a threshold access structure if there exists some positive integer* $t_{\mathbb{A}}$ *such that* $\omega \in \mathbb{A}$ *if and only if* $|w| \geq t_{\mathbb{A}}$*.*

An important notion related to that of access structures in the notion of minimal sets in the access structure.

Definition 3 (Minimal Set). *Given some access structure* $\mathbb{A} \subseteq 2^{\mathcal{P}}$*,* $\omega \in \mathbb{A}$ *is a minimal set if and only if any subset* $\omega' \subset \omega$ *is not authorized, that is,* $\omega' \notin \mathbb{A}$*.*

Finally, we will need the notion of linear secret sharing schemes.

Definition 4 (Linear Secret Sharing Scheme). *Let* $\mathcal{P}$ *be a set of parties. Let* L *be a* $\ell \times k$ *matrix with entries in* $\mathbb{Z}_p$ *together with a one-to-one function* $\pi : \{1, \ldots, \ell\} \to \mathcal{P}$ *which maps a row to a party for labeling. Then* $\Pi = (L, \pi)$ *is a linear secret sharing scheme (LSSS) in* $\mathbb{Z}_p$ *for the access structure* $\mathbb{A}$ *over a set of parties* $\mathcal{P}$ *if, for every set* $\omega \subset \mathcal{P}$*, the vector* $(1, 0, \ldots, 0)$ *is in the* $\mathbb{Z}_p$*-linear span of the rows of* L *labeled by the elements of* ω *if and only if* $\omega \in \mathbb{A}$*. In this case,* $\Pi = (L, \pi)$ *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* $\vec{\beta} = (s, \beta_2, \ldots, \beta_k)^\top$*. It outputs* $L \cdot \vec{\beta}$ *as the vector of* ℓ *shares. The share* $\lambda_i := \langle \vec{L_i}, \vec{\beta} \rangle$ *belongs to party* $\pi(i)$*, where* $\vec{L_i}^\top$ *is the* i^{th} *row of* L*.*

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

An access structure $\mathbb{A}$ is said to be LSSS realizable if there exists a LSSS for $\mathbb{A}$.

2.2 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[7] here and then instantiating to these primitives in the next subsections.

Syntax. Let $R : \Sigma_k \times \Sigma_e \to \{0, 1\}$ be a boolean function where Σ_k and Σ_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 $X \in \Sigma_k$. 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 $Y \in \Sigma_e$. It outputs a ciphertext C.

[7] The term "functional encryption" was defined in slightly different manners in [33, 5, 41] before recently fully formalized in [14]. Our definition of FE here and throughout the paper refers to the class of predicate encryption with public index of [14].

$\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{SK}_X, X, C, Y) \to \mathsf{M}$ 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.

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. Constructions satisfying this security property are sometimes called *payload hiding* in the literature.

A stronger property, called *attribute-hiding*, guarantees that ciphertexts additionally hide their underlying attributes Y and it will not be considered here. To date, this property has only been obtained (e.g., [31]) for access policies that are less expressive than those considered in this paper. We henceforth consider FE systems with public index (according to the terminology of [14]), where ciphertext attributes Y are public.

Definition 5. *A FE scheme for relation R is fully secure (or payload-hiding) 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, the adversary $\mathcal{A}$ chooses a key index X and obtains a private key $\mathsf{SK}_X = \mathsf{Keygen}(\mathsf{msk}, X)$. 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 $\sigma \in \{0, 1\}$, generates a ciphertext $C^\star = \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}_\sigma, Y^\star)$, 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 $\sigma' \in \{0, 1\}$ and wins if $\sigma' = \sigma$.*

The advantage of the adversary $\mathcal{A}$ is measured by $\mathbf{Adv}(\lambda) := |\Pr[\sigma' = \sigma] - \frac{1}{2}|$ *where the probability is taken over all coin tosses.*

A weaker notion called selective security [17, 8] 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 [5], 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 $Y^\star$.

2.3 Ciphertext-Policy Attribute-Based Encryption

In a ciphertext-policy attribute-based encryption scheme, ciphertexts are associated with access structures over the subsets of at most n attributes of the space of attributes, for some specified $n \in \mathbb{N}$. Decryption works only if the attribute set ω associated to a certain secret key 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 6 (CP-ABE). *Let U be an attribute space. Given some $n \in \mathbb{N}$, let $\mathcal{AS}$ be any collection of access structures over U such that for any $\mathbb{A} \in \mathcal{AS}$, there exists some subset $B \subset U$ with $|B| \leq n$ and such that every minimal set ω of $\mathbb{A}$ satisfies that $\omega \subset B$. A ciphertext-policy attribute-based*

encryption (CP-ABE) for the collection $\mathcal{AS}$ is a functional encryption for $R^{\mathsf{CP}} : 2^U \times \mathcal{AS} \to \{0,1\}$ defined by $R^{\mathsf{CP}}(\omega, \mathbb{A}) = 1$ iff $\omega \in \mathbb{A}$ (for any $\omega \subseteq U$ and $\mathbb{A} \in \mathcal{AS}$). Furthermore, the description des *consists of the attribute universe U and the bound n, whereas $\Sigma_k^{\mathsf{CP}} = 2^U$ and $\Sigma_e^{\mathsf{CP}} = \mathcal{AS}$.*

Our construction is only for threshold access structures, i.e. when each access structure $\mathbb{A}$ in the collection $\mathcal{AS}$ is of the threshold type, and admits also some weighted threshold access structures, as we discuss in subsection 3.4.

2.4 Key-Policy Attribute-Based Encryption

Compared with its ciphertext-policy counterpart, in a key-policy attribute-based encryption scheme, access structures and attributes play a dual role. In this case, ciphertexts are associated with a set of attributes ω and private keys correspond to access structures $\mathbb{A}$. Decryption is also only possible when the attribute set ω is authorized in the access structure $\mathbb{A}$. The formal definition as an instance of FE is the following.

Definition 7 (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 and the bound n, whereas $\Sigma_k^{\mathsf{KP}} = \mathcal{AS}$ and $\Sigma_e^{\mathsf{KP}} = \binom{U}{<n}$.*

Definition 7 conforms with the original definition of KP-ABE, as in [28, 36, 32, 33, 14]. There is another variant of KP-ABE recently used in [41], 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 [43].

2.5 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 [32] allows a sender to specify a revoked set S so that only a user with $\mathsf{ID} \notin S$ can decrypt.

Definition 8. *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 9. *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 $S^\star = \tilde{S}$.

2.6 Bilinear Maps and 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 $g, h \neq 1_{\mathbb{G}}$.

The Decision Bilinear Diffie-Hellman Exponent Problem. We rely on the DBDHE assumption introduced in [12]. This assumption is shown to hold in the generic group model [9]. In addition, it is non-interactive and falsifiable [34].

Definition 10. *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 $\mathbb{G}_T$. The advantage of a distinguisher $\mathcal{B}$ is measured by the distance*

$$\begin{aligned}\mathbf{Adv}^{\text{q-DBDHE}}_{\mathbb{G},\mathbb{G}_T}(\lambda) = \big|&\Pr[\mathcal{B}(g, g^\gamma, \ldots, g^{(\gamma^q)}, g^{(\gamma^{q+2})}, \ldots, g^{(\gamma^{2q})}, h, e(g,h)^{(\gamma^{q+1})}) = 1 | \alpha \xleftarrow{R} \mathbb{Z}_p,\ h \xleftarrow{R} \mathbb{G}] \\ &- \Pr[\mathcal{B}(g, g^\gamma, \ldots, g^{(\gamma^q)}, g^{(\gamma^{q+2})}, \ldots, g^{(\gamma^{2q})}, h, T) = 1 | \alpha \xleftarrow{R} \mathbb{Z}_p,\ h \xleftarrow{R} \mathbb{G},\ T \xleftarrow{R} \mathbb{G}_T]\big|\end{aligned}$$

where probabilities are taken over the random choice of g, h, γ, T and the distinguisher $\mathcal{B}$'s coins.

The Augmented Multi-Sequence of Exponents Diffie-Hellman Problem. The security of our first scheme relies on the hardness of a problem that we call the *augmented multi-sequence of exponents decisional Diffie-Hellman problem*, which is a slight modification of the multi-sequence of exponents decisional Diffie-Hellman problem considered in [26]. The generic complexity of these two problems is covered by the analysis in [9], because the problems fit their *general Diffie-Hellman exponent problem* framework.

Let $\tilde{\ell}, \tilde{m}, \tilde{t}$ be three integers. In bilinear groups $(\mathbb{G}, \mathbb{G}_T)$, the *$(\tilde{\ell}, \tilde{m}, \tilde{t})$-augmented multi-sequence of exponents decisional Diffie-Hellman problem* ($(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-DDH) is as follows:

Input: the vector $\overrightarrow{x}_{\tilde{\ell}+\tilde{m}} = (x_1, \ldots, x_{\tilde{\ell}+\tilde{m}})$ whose components are pairwise distinct elements of $\mathbb{Z}_p^\star$ which define the polynomials

$$f(X) = \prod_{i=1}^{\tilde{\ell}} (X + x_i) \qquad \text{and} \qquad g(X) = \prod_{i=\tilde{\ell}+1}^{\tilde{\ell}+\tilde{m}} (X + x_i),$$

the values

$$\begin{cases} g_0, g_0^\gamma, \ldots, g_0^{\gamma^{\tilde{\ell}+\tilde{t}-2}}, & g_0^{\kappa \cdot \gamma \cdot f(\gamma)}, & \text{(l.1)} \\ g_0^{\beta\gamma}, \ldots, g_0^{\beta\gamma^{\tilde{\ell}+\tilde{t}-2}}, & & \text{(l.2)} \\ g_0^\alpha, g_0^{\alpha\gamma}, \ldots, g_0^{\alpha\gamma^{\tilde{\ell}+\tilde{t}}}, & & \text{(l.3)} \\ h_0, h_0^\gamma, \ldots, h_0^{\gamma^{\tilde{m}-2}}, & h_0^{\kappa \cdot g(\gamma)} & \text{(l.4)} \\ h_0^\beta, h_0^{\beta\gamma}, \ldots, h_0^{\beta\gamma^{\tilde{m}-1}}, & & \text{(l.5)} \\ h_0^\alpha, h_0^{\alpha\gamma}, \ldots, h_0^{\alpha\gamma^{2(\tilde{m}-\tilde{t})+3}} & & \text{(l.6)} \end{cases}$$

where $\kappa, \alpha, \gamma, \beta$ are unknown random elements of $\mathbb{Z}_p^\star$, and finally an element $T \in \mathbb{G}_T$.

Output: a bit b.

The problem is correctly solved if the output is $b = 1$ when $T = e(g_0, h_0)^{\kappa \cdot f(\gamma)}$ or if the output is $b = 0$ when T is a random value from $\mathbb{G}_T$. In other words, the goal is to distinguish if T is a random value or if it is equal to $e(g_0, h_0)^{\kappa \cdot f(\gamma)}$.

More formally, let us denote by real the event that T is indeed equal to $T = e(g_0, h_0)^{\kappa \cdot f(\gamma)}$, by random the event that T is a random element from $\mathbb{G}_T$ and by $\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \beta, T)$ the input of the problem. Then, we define the *advantage* of an algorithm $\mathcal{B}$ in solving the $(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-DDH problem as

$$\begin{aligned}\mathbf{Adv}_{\mathcal{B}}^{(\tilde{\ell},\tilde{m},\tilde{t})\text{-aMSE-DDH}}(\lambda) = \Big| &\Pr\left[\mathcal{B}(\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \beta, T)) = 1 \middle| \mathsf{real}\right] \\ &- \Pr\left[\mathcal{B}(\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \beta, T)) = 1 \middle| \mathsf{random}\right] \Big|\end{aligned}$$

where the probability is taken over all random choices and over the random coins of $\mathcal{B}$.

The only difference with the multi-sequence of exponents decisional Diffie-Hellman problem from [26] is the presence in the input of two additional lines (l.2) and (l.5). The generic hardness of this problem is a consequence of Theorem A.2 from [9]. It is stated in the next proposition whose proof follows (almost exactly) that of Corollary 3 in [26].

Proposition 1. *For any probabilistic algorithm $\mathcal{B}$ making at most q_G queries to the the oracle that computes the group operations (in groups $\mathbb{G}, \mathbb{G}_T$ of order p) and the bilinear map $e(\cdot,\cdot)$, its advantage in solving the aMSE-DDH problem satisfies*

$$\mathbf{Adv}_{\mathcal{B}}^{(\tilde{\ell},\tilde{m},\tilde{t})\text{-aMSE-DDH}}(\lambda) \leq \frac{(q_G + 2s + 2)^2 \cdot d}{2p}$$

where $s = 4\tilde{m} + 3\tilde{\ell} + \tilde{t} + 3$ and $d = \max\{2(\tilde{\ell}+2), 2(\tilde{m}+2), 4(\tilde{m}-\tilde{t})+10\}$.

3 A CP-ABE Scheme with Short Ciphertexts for Threshold Policies

This section is dedicated to the presentation of our ciphertext-policy attribute-based encryption scheme, which works for threshold decryption policies.

In the decryption process, we will use the algorithm Aggregate of [25, 26]. Given a list of values $\{g^{\frac{r}{\gamma+x_i}}, x_i\}_{1\leq i\leq n}$, where $r, \gamma \in (\mathbb{Z}_p)^\star$ are unknown and $x_i \neq x_j$ if $i \neq j$, the algorithm computes the value

$$\mathsf{Aggregate}(\{g^{\frac{r}{\gamma+x_i}}, x_i\}_{1\leq i\leq n}) = g^{\frac{r}{\prod_{i=1}^{n}(\gamma+x_i)}}.$$

using $O(n^2)$ exponentiations.

Although the algorithm Aggregate of [25, 26] is given for elements in $\mathbb{G}_T$, it is immediate to see that it works in any group of prime order. Running Aggregate for elements in $\mathbb{G}$ results in our case in a more efficient decryption algorithm.

Concretely, the algorithm proceeds by defining $A_{0,\eta} = g^{r/(\gamma+x_\eta)}$ for each $\eta \in \{1, \ldots, n\}$ and observing that, if we define

$$A_{j,\eta} = g^{\frac{r}{(\gamma+x_\eta)\cdot\prod_{i=1}^{j}(\gamma+x_i)}} \qquad \text{with} \quad 1 \leq j < \eta \leq n,$$

these values satisfy the recursion formula

$$\Lambda_{j,\eta} = \Big(\frac{\Lambda_{j-1,j}}{\Lambda_{j-1,\eta}}\Big)^{1/(x_\eta - x_j)}. \tag{1}$$

Therefore, as long as elements $x_1, \ldots, x_n$ are pairwise distinct, (1) allows sequentially computing $\Lambda_{j,\eta}$ for $j = 1$ to $n-1$ and $\eta = j+1$ to n in order to finally obtain $\Lambda_{n-1,n} = g^{\overline{\prod_{i=1}^{n} (\gamma + x_i)}}$.

3.1 Description

▶ $\mathsf{Setup}(\lambda, U, n)$: the trusted setup algorithm chooses a suitable encoding τ sending each of the m attributes $\mathsf{at} \in U$ onto a (different) element $\tau(\mathsf{at}) = x \in (\mathbb{Z}_p)^\star$. It also chooses groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and generators $g, h \xleftarrow{R} \mathbb{G}$. Then, it chooses a set $\mathcal{D} = \{d_1, \ldots, d_{n-1}\}$ consisting of $n-1$ pairwise different elements of $(\mathbb{Z}_p)^\star$, which must also be different to the values $x = \tau(\mathsf{at})$, for all $\mathsf{at} \in U$. For any integer i lower or equal to $n-1$, we denote as $\mathcal{D}_i$ the set $\{d_1, \ldots, d_i\}$. Next, the algorithm picks at random $\alpha, \gamma \in \mathbb{Z}_p^\star$ and sets $u = g^{\alpha\gamma}$ and $v = e(g^\alpha, h)$. The master secret key is then $\mathsf{msk} = (g, \alpha, \gamma)$ and the public parameters are

$$\mathsf{mpk} = \Big(U, n, u, v, \Big\{h^{\alpha\gamma^i}\Big\}_{i=0,\ldots,2n-1}, \mathcal{D}, \tau\Big).$$

▶ $\mathsf{Keygen}(\mathsf{msk}, \omega)$: to generate a key for the attribute set $\omega \subset U$, pick $r, z \xleftarrow{R} \mathbb{Z}_p^*$ and compute the private key

$$\mathsf{sk}_\omega = \Big(\Big\{g^{\frac{r}{\gamma + \tau(\mathsf{at})}}\Big\}_{\mathsf{at} \in \omega}, \Big\{h^{r\gamma^i}\Big\}_{i=0,\ldots,n-2}, h^{\frac{r-z}{\gamma}}, z\Big).$$

▶ $\mathsf{Encrypt}(\mathsf{mpk}, S, t, \mathsf{M})$: given a subset $S \subset U$ with $s = |S|$ attributes, $s \le n$, a threshold t satisfying $1 \le t \le s$, and a message $\mathsf{M} \in \mathbb{G}_T$, the sender picks at random $\kappa \in \mathbb{Z}_p^\star$ and computes

$$\begin{cases} C_1 = u^{-\kappa}, \\ C_2 = h^{\kappa\cdot\alpha\cdot \prod\limits_{\mathsf{at}\in S} (\gamma + \tau(\mathsf{at}))} \prod\limits_{d \in \mathcal{D}_{n+t-1-s}} (\gamma + d)}, \\ K = v^\kappa = e(g^\alpha, h)^\kappa. \end{cases}$$

The value C_2 is computed from the set $\{h^{\alpha\gamma^i}\}_{i=0,\ldots,2n-1}$ that can be found in the public parameters. The ciphertext is then $C = (C_1, C_2, C_3)$, where $C_3 = K \cdot \mathsf{M}$.

▶ $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_\omega, \omega, C, (S, t))$: given $C = (C_1, C_2, C_3) \in \mathbb{G}^2 \times \mathbb{G}_T$, any user with a set of attributes ω such that $|\omega \cap S| \ge t$ can use the secret key sk_ω to decrypt the ciphertext, as follows. Let ω_S be any subset of $\omega \cap S$ with $|\omega_S| = t$. The user computes, from all $\mathsf{at} \in \omega_S$, the value

$$\mathsf{Aggregate}(\{g^{\frac{r}{\gamma + \tau(\mathsf{at})}}, \tau(\mathsf{at})\}_{\mathsf{at}\in\omega_S}) = g^{\overline{\prod_{\mathsf{at}\in\omega_S} (\gamma + \tau(\mathsf{at}))}}.$$

With the output of the algorithm $\mathsf{Aggregate}$, the decryption algorithm also computes

$$\chi = e(g^{\overline{\prod_{\mathsf{at}\in\omega_S} (\gamma + \tau(\mathsf{at}))}}, C_2) = e(g, h)^{\kappa\cdot\alpha\cdot r\cdot \prod\limits_{\mathsf{at}\in S\setminus\omega_S} (\gamma + \tau(\mathsf{at}))} \prod\limits_{d \in \mathcal{D}_{n+t-1-s}} (\gamma + d)}.$$

For simplicity, let $\tau(d) = d$ for all $d \in \mathcal{D}$ and define $P_{(\omega_S, S)}(\gamma)$ as

$$P_{(\omega_S,S)}(\gamma) = \frac{1}{\gamma}\left(\prod_{y \in (S \cup \mathcal{D}_{n+t-1-s})\setminus\omega_S} (\gamma + \tau(y)) - \prod_{y \in (S \cup \mathcal{D}_{n+t-1-s})\setminus\omega_S} \tau(y)\right).$$

The crucial point is that, since $|\omega_S| \geq t$, the degree of the polynomial $P_{(\omega_S,S)}(X)$ is lower or equal to $n-2$. Therefore, from the values included in sk_ω, the user can compute $h^{rP_{(\omega_S,S)}(\gamma)}$.
After that, the user calculates

$$e(C_1, h^{rP_{(\omega_S,S)}(\gamma)}) \cdot \chi = e(g,h)^{\kappa\cdot\alpha\cdot r\cdot\prod_{y\in(S\cup\mathcal{D}_{n+t-1-s})\setminus\omega_S}\tau(y)} \tag{2}$$

and

$$e(C_1, h^{\frac{r-z}{\gamma}}) = e(g,h)^{-\kappa\cdot\alpha\cdot r}\cdot e(g,h)^{\kappa\cdot\alpha\cdot z} \tag{3}$$

From Equation (2), the decryption algorithm obtains

$$e(g,h)^{\kappa\cdot\alpha\cdot r} = \left(e(C_1, h^{rP_{(\omega_S,S)}(\gamma)})\cdot\chi\right)^{1/\prod_{y\in(S\cup\mathcal{D}_{n+t-1-s})\setminus\omega_S}\tau(y)}$$

and multiplies this value in Equation (3). The result of this multiplication leads to $e(g,h)^{\kappa\cdot\alpha\cdot z}$. This value is raised to z^{-1} to obtain $K = e(g,h)^{\kappa\cdot\alpha}$. Finally, the plaintext is recovered by computing $\mathsf{M} = C_3/K$.

3.2 Consistency Checking and Efficiency Considerations

It is not hard to prove that the new ABE scheme satisfy the correctness property: if all the protocols are correctly executed, and if $|\omega \cap S| \geq t$, then sk_ω allows recovering plaintexts that have been encrypted for the pair (S,t).

It is worth noting that, by adding g^α to the public parameters (this modification does not affect the security proof that we present in the next section), the users can check the consistency of the secret key they receive from the master entity. To do so, they must verify that, for all their attributes $\mathsf{at} \in \omega$,

$$e\left(g^{\frac{r}{\gamma+\tau(\mathsf{at})}}, h^{\alpha\gamma}\cdot(h^\alpha)^{\tau(\mathsf{at})}\right) = e\left(g^\alpha, h^r\right)$$

and then that, for $i = 1,\ldots,n-2$,

$$e\left(g^\alpha, h^{r\gamma^i}\right) = e\left(u, h^{r\gamma^{i-1}}\right)$$

Finally, they have to check that $e(u, h^{\frac{r-z}{\gamma}}) = e\left(g^\alpha, h^r\right)/v^z$.

In terms of efficiency, the main contribution of this new scheme is the constant size of the ciphertext, which consists of two elements of $\mathbb{G}$ and one element of $\mathbb{G}_T$. Encryption requires no pairing computations, but $n+t+1$ exponentiations. The decryption process requires 3 pairing evaluations and $O(t^2+n)$ exponentiations. The size of the secret key is linear in the number of attributes, as in all existing ABE schemes.

3.3 Security

We are going to prove that our scheme is selectively secure, assuming that the aMSE-DDH problem is hard to solve.

Theorem 1. *Let λ be an integer. For any adversary $\mathcal{A}$ against the selective security of our CP-ABE scheme, for a universe U of m attributes and maximal size $n \geq |\tilde{S}|$ for any decryption policy $(\tilde{S},\tilde{t})$, there exists a solver $\mathcal{B}$ of the $(\tilde{\ell},\tilde{m},\tilde{t})$-aMSE-DDH problem such that*

$$\mathbf{Adv}_{\mathcal{B}}^{\text{aMSE-DDH}}(\lambda) \;\geq\; \frac{1}{n^2}\cdot\mathbf{Adv}_{\mathcal{A}}^{\text{ABE-sCPA}}(\lambda).$$

Proof. We are going to construct an algorithm $\mathcal{B}$ that uses the adversary $\mathcal{A}$ as a black-box and that solves an instance of the aMSE-DDH problem. The main trick in the proof will be to use the input of the aMSE-DDH problem to compute evaluations of some polynomials in γ "in the exponent".

$\mathcal{B}$ chooses values m, n for the size of the universe of attributes $U = \{\mathsf{at}_1, \ldots, \mathsf{at}_m\}$ and for the upper bond on the size of allowed decryption policies. After that, $\mathcal{B}$ chooses at random two integers s, t such that $1 \leq t \leq s \leq n$. Then, $\mathcal{B}$ asks for an instance of the $(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-DDH problem, where $\tilde{\ell} = m - s$, $\tilde{m} = n + t - 1$ and $\tilde{t} = t + 1$.

Let $\mathcal{I}(\overrightarrow{x}_{m+n+t-1-s}, \kappa, \alpha, \gamma, \beta, T)$ be the instance of the problem received by $\mathcal{B}$. Now, $\mathcal{B}$ initializes the adversary $\mathcal{A}$ against the selective security of the CP-ABE scheme. The adversary $\mathcal{A}$ chooses a set $S \subset U$ of cardinal s' that he wants to attack, and a threshold t' such that $1 \leq t' \leq s' \leq n$. If $(s', t') \neq (s, t)$, then $\mathcal{B}$ aborts and outputs a random bit as the answer to the aMSE-DDH problem.

Otherwise (that is, if $s' = s$ and $t' = t$, which happens with probability at least $1/n^2$), the solver $\mathcal{B}$ goes on with the simulation of the environment of adversary $\mathcal{A}$.

Without loss of generality, we assume $S = \{\mathsf{at}_{m-s+1}, \ldots, \mathsf{at}_m\} \subset U$. From now on, we will denote by ω_S the subset $\omega \cap S$, for any subset of attributes ω.

Simulation of the setup phase. Algorithm $\mathcal{B}$ defines the encoding of the attributes as $\tau(\mathsf{at}_i) = x_i$ for $i = 1, \ldots, m$. Observe that the encodings of the first $m - s$ elements are the opposite of the roots of $f(X)$, and the encodings of the attributes in S are the opposite of some roots of $g(X)$.

The values corresponding to "dummy" attributes $\mathcal{D} = \{d_1, \ldots, d_{n-1}\}$ are defined as $d_j = x_{m+j}$ if $j = 1, \ldots, n + t - 1 - s$. For $j = n + t - s, \ldots, n - 1$, the d_j's are picked uniformly at random in $\mathbb{Z}_p^\star$ until they are distinct from $\{x_1, \ldots, x_{m+n+t-1-s}, d_{n+t-s}, \ldots, d_{j-1}\}$.

Our algorithm $\mathcal{B}$ defines $g := g_0^{f(\gamma)}$. Note that $\mathcal{B}$ can compute g with the elements of line (l.1) of its input, since f is a polynomial of degree $\tilde{\ell}$. To complete the setup phase, $\mathcal{B}$ sets $h = h_0$ and computes

- $u = g^{\alpha\gamma} = g_0^{\alpha \cdot \gamma \cdot f(\gamma)}$ with line (l.3) of its input, which is possible since $Xf(X)$ is a polynomial of degree $\tilde{\ell} + 1$. Indeed, $\alpha \cdot \gamma \cdot f(\gamma)$ is a linear combination of $\{\alpha\gamma, \ldots, \alpha\gamma^{\tilde{\ell}+1}\}$ and the coefficients of this linear combination are known to $\mathcal{B}$, so the value u can be computed from line (l.3).
- $v = e(g, h)^\alpha = e(g_0^{f(\gamma)\alpha}, h_0)$ with line (l.3) for $g_0^{f(\gamma)\alpha}$. Note that the value g^α could be computed by $\mathcal{B}$ and added to the public parameters, in case the verification of the consistency of the secret keys is desired for the scheme.

Algorithm $\mathcal{B}$ can compute the values $\{h^{\alpha\gamma^i}\}_{i=0,\ldots,2n-1}$ from line (l.6) of its input. Eventually, $\mathcal{B}$ provides $\mathcal{A}$ with the resulting master public key

$$\mathsf{mpk} = \Big(U, n, u, v, \{h^{\alpha\gamma^i}\}_{i=0,\ldots,2n-1}, \mathcal{D}, \tau\Big).$$

Simulation of key extraction queries. Whenever the adversary $\mathcal{A}$ makes a key extraction query for a subset of attributes $\omega \subset U$ satisfying that $0 \leq |\omega_S| \leq t - 1$, the algorithm $\mathcal{B}$ must produce a tuple of the form

$$\mathsf{sk}_\omega = \Big(\Big\{ g^{\frac{r}{\gamma + \tau(\mathsf{at})}} \Big\}_{\mathsf{at} \in \omega}, \Big\{ h^{r\gamma^i} \Big\}_{i=0,\ldots,n-2}, h^{\frac{r-z}{\gamma}}, z \Big),$$

for some random values $r, z \in \mathbb{Z}_p^\star$. To do so, $\mathcal{B}$ chooses $z \in \mathbb{Z}_p^\star$ uniformly at random and implicitly defines $r = (\beta \cdot y_\omega \cdot \gamma + z) \cdot Q_\omega(\gamma)$, where y_ω is randomly picked in $\mathbb{Z}_p^\star$, and the polynomial $Q_\omega(X)$ is defined as $Q_\omega(\gamma) = 1$ when $|\omega_S| = 0$, or $Q_\omega(X) = \lambda_\omega \cdot \prod_{\mathsf{at} \in \omega_S} (X + \tau(\mathsf{at}))$ otherwise, in which case $\lambda_\omega = (\prod_{\mathsf{at} \in \omega_S} \tau(\mathsf{at}))^{-1}$.

The rest of elements (other than z) which form sk_ω are then computed as follows:

– For any $\mathsf{at} \in \omega_S$, $\mathcal{B}$ defines

$$Q_{\mathsf{at}}(\gamma) = Q_\omega(\gamma)/(\gamma + \tau(\mathsf{at})) = \lambda_\omega \cdot \prod_{\tilde{\mathsf{at}} \in \omega_S,\ \tilde{\mathsf{at}} \neq \mathsf{at}} (\gamma + \tau(\tilde{\mathsf{at}})).$$

Then $g^{\frac{r}{\gamma+\tau(\mathsf{at})}} = g_0^{f(\gamma)\cdot\beta\cdot y_\omega\cdot\gamma\cdot Q_{\mathsf{at}}(\gamma)} \cdot g_0^{z\cdot f(\gamma)\cdot Q_{\mathsf{at}}(\gamma)}$. The first factor of the product (whose exponent is a polynomial in γ of degree at most $(m-s)+1+t-2$) can be computed from line (l.2), whereas the second factor (whose exponent is a polynomial in γ of degree at most $(m-s)+t-2$) can be computed from line (l.1).

– For any attribute $\mathsf{at} \in \omega \backslash \omega_S$, $\mathcal{B}$ defines the polynomial $f_{\mathsf{at}}(X) = f(X)/(X+\tau(\mathsf{at}))$ and considers the product $g^{\frac{r}{\gamma+\tau(\mathsf{at})}} = g_0^{f_{\mathsf{at}}(\gamma)\cdot\beta\cdot y_\omega\cdot\gamma\cdot Q_\omega(\gamma)} \cdot g_0^{z\cdot f_{\mathsf{at}}(\gamma)\cdot Q_\omega(\gamma)}$. Again, the first factor of this product can be computed from line (l.2), and the second factor can be computed from line (l.1).

– The values $\left\{h^{r\gamma^i}\right\}_{i=0,\ldots,n-2}$ can be computed from line (l.4) and (l.5), since

$$h^{r\gamma^i} = h^{Q_\omega(\gamma)\cdot\beta\cdot y_\omega\cdot\gamma^{i+1}} \cdot h^{z\cdot Q_\omega(\gamma)\cdot\gamma^i}.$$

– Finally, $\mathcal{B}$ has to compute $h^{\frac{r-z}{\gamma}} = h^{Q_\omega(\gamma)\cdot\beta\cdot y_\omega} \cdot h^{\frac{z\cdot Q_\omega(\gamma)-z}{\gamma}}$. The first factor of the product can be computed from line (l.5) and the second factor can be computed from line (l.4), since by definition of λ_ω, $Q_\omega(X)$ is a polynomial with independent term equal to 1 and thus $\frac{zQ_\omega(\gamma)-z}{\gamma}$ is a linear combination of $\{1, \gamma, \ldots, \gamma^{t-2}\}$.

Note that $Q_\omega(\gamma) \neq 0$ (otherwise $\gamma = \tau(at)$ for some $\mathsf{at} \in \omega_S$ and γ is public), in which case it is not hard to see that r is uniformly distributed in $\mathbb{Z}_p$. If the choice of y_ω leads to $r = 0$ (which occurs only with negligible probability anyhow), it suffices to pick a different value for y_ω. That is, in the simulation r is uniformly distributed in $\mathbb{Z}_p^\star$.

Challenge. Once $\mathcal{A}$ sends to $\mathcal{B}$ the two messages M_0 and M_1, $\mathcal{B}$ flips a coin $\sigma \xleftarrow{R} \{0,1\}$ and sets $C_3^\star = T \cdot \mathsf{M}_\sigma$. To simulate the rest of the challenge ciphertext, $\mathcal{B}$ implicitly defines the randomness for the encryption as $\kappa' = \kappa/\alpha$, and sets $C_2^\star = h_0^{\kappa\cdot g(\gamma)}$ (given in line (l.4) of the aMSE-DDH input). To complete the generation of the challenge ciphertext, $\mathcal{B}$ computes $C_1^\star = \left(g_0^{\kappa\cdot\gamma f(\gamma)}\right)^{-1}$ from (l.1) of the input, which is equal to $u^{-\kappa'}$.

After the challenge step $\mathcal{A}$ may make other key extraction queries, which are answered as before.

Guess. Finally, $\mathcal{A}$ outputs a bit σ'. If $\sigma' = \sigma$, $\mathcal{B}$ answers 1 as the solution to the given instance of the aMSE-DDH problem, meaning that $T = e(g_0, h_0)^{\kappa\cdot f(\gamma)}$. Otherwise, $\mathcal{B}$ answers 0, meaning that T is a random element.

We now have to analyze the advantage of the distinguisher $\mathcal{B}$. If $\mathcal{B}$'s guess of the values s and t was always correct, then we would have:

$$\begin{aligned}\mathbf{Adv}_{\mathcal{B}}^{\text{aMSE-DDH}}(\lambda) &= \Big|\Pr\left[\mathcal{B}(\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \beta, T)) = 1 \middle| \mathsf{real}\right] - \\ &\qquad \Pr\left[\mathcal{B}(\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \beta, T)) = 1 \middle| \mathsf{random}\right]\Big| \\ &= \Big|\Pr\left[\sigma = \sigma' \middle| \mathsf{real}\right] - \Pr\left[\sigma = \sigma' \middle| \mathsf{random}\right]\Big|.\end{aligned}$$

When real occurs, $\mathcal{A}$ is playing a real attack and we have $|\Pr\left[\sigma = \sigma' \middle| \mathsf{real}\right] - 1/2| = \mathbf{Adv}_{\mathcal{A},\Pi}^{\text{IND-sCPA}}(\lambda)$. During the random event, $\mathcal{A}$'s view is completely independent of the bit $\sigma \in \{0,1\}$, so that we have

$\Pr[\sigma = \sigma' | \mathsf{random}] = 1/2$. Combining these arguments with the fact that $\mathcal{B}$'s guess of the values s and t is actually correct with some probability at least $1/n^2$, we obtain

$$\mathbf{Adv}_{\mathcal{B}}^{\text{aMSE-DDH}}(\lambda) \;\geq\; \frac{1}{n^2} \cdot \mathbf{Adv}_{\mathcal{A},\Pi}^{\text{ABE-sCPA}}(\lambda).$$

□

3.4 More General Decryption Policies

Although we have considered in this paper the special case of threshold decryption policies, attribute-based encryption schemes can be defined for general decryption policies. Such a policy is determined by a monotone increasing family $\mathbb{A} \subset 2^U$ of subsets of attributes, in $U = \{\mathsf{at}_1, \ldots, \mathsf{at}_m\}$. This family (or *access structure*) is chosen by the sender at the time of encryption, in such a way that only users whose subset of attributes ω belong to $\mathbb{A}$ can decrypt. Even if many users collude, each of them having a subset of attributes out of $\mathbb{A}$, the encryption scheme must remain secure.

The threshold ABE scheme that we have described and analyzed in this paper is inspired on the dynamic threshold identity-based encryption scheme of [26]. It is claimed in [26] that the threshold scheme there can be extended to admit "all the classical cases" of more general access structures. However, this is not completely true, because their extension only applies to a sub family of access structures, *weighted threshold* ones. A family $\mathbb{A} \subset 2^U$ is a weighted threshold access structure if there exist a threshold t and an assignment of weights $\mathrm{wt} : U \to \mathbb{Z}^+$ such that $\omega \in \mathbb{A} \iff \sum_{\mathsf{at} \in \omega} \mathrm{wt}(\mathsf{at}) \geq t$. Of course, there are many access structures which are not weighted threshold, for example $\mathbb{A} = \{\{\mathsf{at}_1, \mathsf{at}_2\}, \{\mathsf{at}_2, \mathsf{at}_3\}, \{\mathsf{at}_3, \mathsf{at}_4\}\}$ in the set $U = \{\mathsf{at}_1, \mathsf{at}_2, \mathsf{at}_3, \mathsf{at}_4\}$.

The same extension proposed in [26] works for our threshold ABE scheme. Let K be an upper bound for $\mathrm{wt}(\mathsf{at})$, for all $\mathsf{at} \in U$ and for all possible assignments of weights that realize weighted threshold decryption policies. During the setup of the ABE scheme, the new universe of attributes will be $U' = \{\mathsf{at}_1||1, \mathsf{at}_1||2, \ldots, \mathsf{at}_1||K, \ldots, \mathsf{at}_m||1, \ldots, \mathsf{at}_m||K\}$. During the secret key request phase, if an attribute at belongs to the requested subset $\omega \subset U$, the secret key sk_ω will contain the elements $g^{\frac{r}{\gamma + \tau(\mathsf{at}^{(j)})}}$ corresponding to $\mathsf{at}^{(j)} = \mathsf{at}||j$, for all $j = 1, \ldots, K$.

Later, suppose a sender wants to encrypt a message for a weighted threshold decryption policy $\mathbb{A}$, defined on a subset of attributes $S = \{\mathsf{at}_1, \ldots, \mathsf{at}_s\}$ (without loss of generality). Let t and $\mathrm{wt} : S \to \mathbb{Z}^+$ be the threshold and assignment of weights that realize $\mathbb{A}$. The sender can use the threshold ABE encryption routine described in Section 3.1, with threshold t, but applied to the set of attributes $S' = \{\mathsf{at}_1||1, \ldots, \mathsf{at}_1||\mathrm{wt}(\mathsf{at}_1), \ldots, \mathsf{at}_s||1, \ldots, \mathsf{at}_s||\mathrm{wt}(\mathsf{at}_s)\}$. In this way, if a user holds a subset of attributes $\omega \in \mathbb{A}$, he will have $\mathrm{wt}(\mathsf{at})$ valid elements in his secret key, for each attribute $\mathsf{at} \in \omega$. In total, he will have $\sum_{\mathsf{at} \in \omega} \mathrm{wt}(\mathsf{at}) \geq t$ valid elements, so he will be able to run the decryption routine of the threshold ABE scheme and decrypt the ciphertext.

The security analysis can be extended to this more general case, as well. Therefore, we can conclude that our ABE scheme with constant size ciphertexts also admits weighted threshold decryption policies.

3.5 Delegation of Secret Keys and Security under Chosen Ciphertext Attacks

Our attribute-based encryption scheme admits delegation of secret keys: from a valid secret key $\mathsf{sk}_\omega = \left(\left\{ g^{\frac{r}{\gamma + \tau(\mathsf{at})}} \right\}_{\mathsf{at} \in \omega}, \left\{ h^{r\gamma^i} \right\}_{i=0,\ldots,n-2}, h^{\frac{r-z}{\gamma}}, z \right)$ it is possible to compute a valid secret key $\mathsf{sk}_{\omega'}$ for any subset $\omega' \subset \omega$, as follows: take $\rho \in \mathbb{Z}_p^\star$ at random and compute

$$\mathsf{sk}_{\omega'} = \left(\left\{ \left(g^{\frac{r}{\gamma + \tau(\mathsf{at})}} \right)^\rho \right\}_{\mathsf{at} \in \omega'}, \left\{ \left(h^{r\gamma^i} \right)^\rho \right\}_{i=0,\ldots,n-2}, \left(h^{\frac{r-z}{\gamma}} \right)^\rho, z \cdot \rho \right).$$

Our ABE scheme can be therefore viewed as a hierarchical ABE scheme with the natural hierarchy: a user holding attributes ω is over a user holding attributes ω', if $\omega' \subset \omega$. Then, the techniques developed in [18] can be applied to transform our hierarchical ABE scheme, which enjoys selective security under chosen plaintext attacks, into an ABE scheme which enjoys selective security under chosen ciphertext attacks, in the standard model. The price to pay is an increase in the size of the secret keys sk_ω, that must contain $2l$ additional elements, where l is the bit-length of the verification keys of a (one-time) signature scheme that is used in the transformation. The size of the ciphertexts remains constant.

4 KP-ABE Constructions with Short Ciphertexts for Monotonic LSSS-Realizable Access Structures

In this section, our goal is to construct monotonic KP-ABE systems with short ciphertexts for any LSSS-realizable access structure. 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.

In spirit, the construction is somewhat similar to the one described by Boyen [15], which transforms IBE systems in the exponent-inversion framework (e.g., [37]) into ABE primitives. The approach of [15] takes 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 [15], our transformation preserves the ciphertext size, so that using IBBE schemes with short ciphertexts yields KP-ABE constructions with the same ciphertext size.

4.1 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 parameters $n, n_1, n_2 \in \mathbb{N}$, an efficiently samplable family $\mathcal{F}$ of vectors (f_1, f_2, F) of functions, and a function $\mathcal{D}$, of which the latter two are specified by

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

with requirements specified below. We assume w.l.o.g. that identities are encoded as elements of $\mathbb{Z}_p^*$ (otherwise, they can always be hashed modulo p) and the linear IBBE template is as follows.

▶ $\mathsf{Setup}(\lambda, n)$: Given a security parameter $\lambda \in \mathbb{N}$ and a strict upper 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 generator $g \xleftarrow{R} \mathbb{G}$. 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 $\mathsf{SK}_{\mathsf{ID}}$ as $(d_1, d_2, d_3) \in \mathbb{G}^{n_1+2}$ and the ciphertext C as $(C_0, C_1, C_2) \in \mathbb{G}_T \times \mathbb{G}^{n_2+1}$. Then, it 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 $\mathsf{M} = C_0 / e(g,g)^{\alpha \cdot s}$.

Moreover, for all $(f_1, f_2, F) \in \mathcal{F}$, the two following 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 $(C_1, C_2) \in \mathbb{G}^{\leq n_2+1}$, we have

$$\mathcal{D}\Big((d_1, d_2, d_3)^\gamma, \mathsf{ID}, (C_1, C_2), S\Big) = \mathcal{D}\Big((d_1, d_2, d_3), \mathsf{ID}, (C_1, C_2), S\Big)^\gamma. \tag{4}$$

4.2 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 $\vec{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)^\top \xleftarrow{R} (\mathbb{Z}_p)^k$ subject to the constraint $\beta_1 = 1$. Then for each $i = 1$ to ℓ, it calculates $\lambda_i = \langle \vec{L_i}, \vec{\beta} \rangle$, picks $r' \xleftarrow{R} \mathbb{Z}_p$ and sets D_i as follows.

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

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

$$\mathcal{D}\big((d'_{i,1}, d'_{i,2}, d'_{i,3}), \mathsf{ID}, (C_1, C_2), S\big) \to e(g,g)^{\alpha \cdot s \cdot \lambda_i}, \tag{5}$$

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 $\sum_{i \in I} \mu_i \cdot \lambda_i = 1$.

CORRECTNESS. We now verify that equation (5) is correct. First, the distribution of private keys in the linear IBBE template is such 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, by construction, we can write

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

where $\tilde{r}_i = r_i + r'/\lambda_i$ and $(d_1, d_2, d_3) = \mathsf{SK}_{\pi(i)}$ denotes an IBBE private key for the identity $\pi(i)$ and the random exponent $\tilde{r}_i$. The linearity requirement (4) then implies

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

for each $i \in I$, which guarantees correctness.

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

Theorem 2. *If the underlying IBBE scheme is selectively secure, then so is the resulting KP-ABE system. More precisely, for any selective-set adversary $\mathcal{A}$ against the KP-ABE construction, there is an IND-sID-CPA adversary $\mathcal{B}$ against the IBBE scheme and*

$$\mathbf{Adv}_{\mathcal{B}}^{\text{IBBE-sID-CPA}}(\lambda) \geq \mathbf{Adv}_{\mathcal{A}}^{\text{KP-ABE-sCPA}}(\lambda).$$

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

The game begins with the KP-ABE adversary $\mathcal{A}$ choosing an attribute set $\omega^\star$ that he intends to attack. The IBBE adversary $\mathcal{B}$ then announces $S^\star = \{i \in \omega^\star\}$ as his target set of receivers. The system-wide IBBE public key that $\mathcal{B}$ receives from his 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 $\vec{1} = (1, 0, \ldots, 0)^\top$ is not in the row space of $L_{\omega^\star}$, there must exist an efficiently computable vector $\vec{w}$ such that $L_{\omega^\star} \cdot \vec{w} = \vec{0}$ and $\langle \vec{1}, \vec{w} \rangle \neq 0$ (according to proposition 1 in [28]). Let h denote the value $\langle \vec{1}, \vec{w} \rangle$. To construct a private key, $\mathcal{B}$ has to define a vector $\vec{u} = \alpha \cdot \vec{\beta}$ such that $\langle \vec{1}, \vec{u} \rangle = \alpha$, which will be used to define shares $\lambda_i = \langle \vec{L}_i, \vec{u} \rangle$. As in the proof of theorem 3 in [28], $\mathcal{B}$ implicitly sets $\vec{u}$ as $\vec{u} = \vec{v} + \psi \cdot \vec{w}$, where $\vec{v} = (v_1, \ldots, v_k)^\top$ is a randomly chosen vector and $\psi = (\alpha - v_1)/h$, so that $\langle \vec{1}, \vec{u} \rangle = \alpha$. 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 $\vec{L}_j^\top = (m_{j1}, \ldots, m_{jk})$ denotes the j^{th} row of L, we have $\langle \vec{L}_j, \vec{w} \rangle = 0$, so that $\langle \vec{L}_j, \vec{u} \rangle = \langle \vec{L}_j, \vec{v} \rangle = \sum_{t_1=1}^{k} m_{jt_1} v_{t_1}$ and the share $\lambda_j = \langle \vec{L}_j, \vec{u} \rangle$ is thus computable, so that $\mathcal{B}$ can pick integers $\lambda_j, r_j \xleftarrow{R} \mathbb{Z}_p^*$ 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 $\Gamma_2 = \{j \in \{1, \ldots, \ell\} \mid \pi(j) \notin \omega^\star\}$. For each $j \in \Gamma_2$, $\mathcal{B}$ is allowed to query its own challenger to extract $(d_{j,1}, d_{j,2}, d_{j,3}) \leftarrow \Pi_{\mathsf{IBBE}}.\mathsf{Keygen}(\mathsf{msk}, \pi(j))$. Also, we have

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

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, he outputs messages $\mathsf{M}_0, \mathsf{M}_1$ that $\mathcal{B}$ forwards to his 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 $\sigma' \in \{0, 1\}$ as $\mathcal{A}$ does. It is easy to see that $\mathcal{B}$ never has to query his 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 and the announced result follows. □

Instantiation Example. The large-universe construction of KP-ABE in [28] 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 immediately deduced from [28]. Since the size of an output from F is linear, ciphertexts in the KP-ABE of [28] are also of linear size.

4.3 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 a functional encryption system with public index – which itself is implied by the spatial encryption scheme of [13] – that was proposed in [5, Sect.4.1] for the zero evaluation of inner-products. Such a FE system is defined by a relation $R^{\mathsf{ZIP}} : \mathbb{Z}_p \times \mathbb{Z}_p \to \{0, 1\}$ where $R^{\mathsf{ZIP}}(\vec{X}, \vec{Y}) = 1$ iff $\langle \vec{X}, \vec{Y} \rangle = 0$. The technique of deriving an IBBE scheme from the zero evaluation of inner-products can be traced to the work of Katz, Sahai and Waters [31]. A private key for an identity ID is defined by setting $\vec{X} = (x_1, \ldots, x_n)^\top$, with $x_i = \mathsf{ID}^{i-1}$. To encrypt to a set $S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_q\}$, one defines $\vec{Y} = (y_1, \ldots, y_n)^\top$ as a coefficient vector from

$$P_S[Z] = \sum_{i=1}^{q+1} y_i Z^{i-1} = \prod_{\mathsf{ID}_j \in S} (Z - \mathsf{ID}_j), \tag{6}$$

where, if $q + 1 < n$, the coordinates $y_{q+2}, \ldots, y_n$ are all set to 0. By doing so, we note that $P_S[\mathsf{ID}] = \langle \vec{X}, \vec{Y} \rangle$ evaluates to 0 iff $\mathsf{ID} \in S$. We now describe the IBBE instantiated from the FE system of [5]. Its selective security is an immediate consequence of [5], 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 $\alpha, \alpha_0 \xleftarrow{R} \mathbb{Z}_p$, $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \xleftarrow{R} \mathbb{Z}_p^n$. It then sets $\vec{H} = (h_1, \ldots, h_n)^\top = g^{\vec{\alpha}}$. The master secret key is $\mathsf{msk} = \alpha$, and the public key is $\mathsf{mpk} = \big(g,\ e(g,g)^\alpha,\ h_0 = g^{\alpha_0},\ \vec{H} = g^{\vec{\alpha}} \big)$.

▶ $\mathsf{Keygen}(\mathsf{msk}, \mathsf{ID})$: The algorithm first defines a vector $\vec{X} = (x_1, \ldots, x_n)^\top$ 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

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

▶ $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, S)$: To encrypt M to the receiver set S (where $|S| < n$), the algorithm defines $\vec{Y} = (y_1, \ldots, y_n)^\top$ as the coefficient vector of $P_S[Z]$ from equation (6). It then picks $s \xleftarrow{R} \mathbb{Z}_p$ and computes the ciphertext as

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

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

$$e(g,g)^{\alpha \cdot s} = \frac{e(C_1, D_1 \cdot K_2^{y_2} \cdots K_n^{y_n})}{e(C_2, D_2)}, \tag{7}$$

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

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

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

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 form

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

where $h_0, h_1, \ldots, h_n \in_R \mathbb{G}$ and the vector $\vec{Y} = (y_1, \ldots, y_n)^\top$ is defined from the polynomial $P_S[Z]$ in equation (6) as usual. In addition, the function $\mathcal{D}$ performs the computation in equation (7), which is easily seen to have linearity, as required.

The resulting KP-ABE construction has constant-size ciphertexts since $n_2 = 1$. This comes at 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 [13] (which is recalled in appendix B) since it also falls into our framework and thus produces an equally efficient KP-ABE scheme.

5 A Scheme Supporting Non-Monotonic Access Structures

Our final goal in this paper is to construct KP-ABE with non-monotonic structures. To this end, we will combine the monotonic KP-ABE system implied by the generic construction of Section 4 with a new revocation mechanism which is presented in Section 5.1.

To securely integrate both schemes, we have to craft the revocation system in such a way that both constructions share some similar structure and rely on the same underlying hard problem. By doing so, the security proof of the resulting non-monotonic KP-ABE scheme goes through and rests on the same assumption as its two components.

5.1 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 [5, 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 4.3 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 [5] but it provides a better efficiency and relies on the DBDHE assumption.

The scheme is a functional encryption system with public index where ciphertexts and private keys both correspond to vectors (say $\vec{Y}$ and $\vec{X}$) and decryption succeeds whenever $\langle \vec{X}, \vec{Y} \rangle \neq \vec{0}$. Such a primitive is easily seen to imply identity-based revocation via the same identity encoding as in section 4.3.

In its simplified version (which uses bilinear groups of composite order groups), the scheme of [5][Appendix E] assigns to a vector $\vec{X} = (x_1, \ldots, x_n)$ a private key of the form

$$(D_1, D_2, \{K_i\}_{i=1}^n) = \big(g^\alpha \cdot h_0^r, g^r, \{(h_0^{x_i} \cdot h_i)^r\}_{i=1}^n\big)$$

whereas ciphertexts for a vector $\vec{Y}$ are encrypted as $(C_0, C_1, C_2) = \big(\mathsf{M} \cdot e(g,g)^{\alpha \cdot s}, g^s, (h_1^{y_1} \cdots h_n^{y_n})^s\big)$. Decryption proceeds by first using $\vec{Y} = (y_1, \ldots, y_n)$ and private key elements $\{K_i\}_{i=1}^n$ to compute $K = \prod_{i=1}^n K_i^{y_i} = (h_0^{\vec{X} \cdot \vec{Y}} \cdot \prod_{i=1}^n h_i^{y_i})^r$, so that $e(K, C_1)/e(C_2, D_2)$ equals $e(h_0, g)^{rs \cdot \vec{X} \cdot \vec{Y}}$. Then, given that $e(D_1, C_1) = e(g,g)^{\alpha \cdot s} \cdot e(h_0, g)^{rs}$, as long as $\langle \vec{X}, \vec{Y} \rangle \neq \vec{0}$, the receiver is able to compute

$$e(g,g)^{\alpha \cdot s} = e(D_1, C_1) \cdot \big(e(K, C_1)/e(C_2, D_2)\big)^{-1/(\vec{X} \cdot \vec{Y})}.$$

The co-selective security of the above scheme was proved in [5] in groups of composite order. In this section, we slightly modify it to prove its co-selective security in prime-order groups under the q-DBDHE assumption. This makes it easier to combine with the concrete monotonic KP-ABE suggested in section 4.3.

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

► $\mathsf{Keygen}(\mathsf{msk}, \mathsf{ID})$: The algorithm first defines a vector $\vec{X} = (x_1, \ldots, x_n)^\top$ 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

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

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

► $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, S)$: To encrypt M with the revoked set S (where $|S| < n$), the algorithm defines $\vec{Y} = (y_1, \ldots, y_n)^\top$ as the coefficient vector of $P_S[Z]$ from equation (6). It then picks $s \xleftarrow{R} \mathbb{Z}_p$ and computes the ciphertext as

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

▶ $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{ID}, C, S)$: It defines $\vec{X}$ from ID and $\vec{Y}$ from S as usual. It then successively computes elements $K = \prod_{i=2}^{n} K_i^{y_i} = \big(h_1^{-\langle \vec{X}, \vec{Y} \rangle / x_1} \cdot h_1^{y_1} \cdots h_n^{y_n}\big)^r$,

$$\chi = \Big(\frac{e\big(K, C_1\big)}{e(C_2, D_2)}\Big)^{-\frac{x_1}{\langle \vec{X}, \vec{Y} \rangle}} = e(g, h_1)^{rs},$$

and then obtains $\mathsf{M} = C_0 \cdot e(C_1, D_1)^{-1} \cdot \chi$.

Correctness. We first observe that

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

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

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

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

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

As already mentioned, this IBR scheme shares the same high-level structure (including the form of the public key and the ciphertext) as the IBBE in section 4.3 and relies on the same assumption. These similarities make it possible to assemble both constructions in the design of a non-monotonic ABE system in Section 5.2.

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 3. *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 $(\mathbb{G}, \mathbb{G}_T)$. Namely, any co-selective adversary $\mathcal{A}$ implies a n-DBDHE distinguisher $\mathcal{B}$ such that*

$$\mathbf{Adv}_{\mathcal{B}}^{n\text{-DBDHE}}(\lambda) \geq \mathbf{Adv}_{\mathcal{A}}^{\text{IBR-co-sCPA}}(\lambda).$$

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 he wishes to obtain private keys. Let $\vec{X}_1, \ldots, \vec{X}_q$ the corresponding vectors. That is, $\vec{X}_k = (1, \mathsf{ID}_k, \mathsf{ID}_k^2, \ldots, \mathsf{ID}_k^{n-1})$. 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 $\vec{H} = (h_1, \ldots, h_n)^{\top}$ are then defined as follows. For each $k \in [1, q]$, $\mathcal{B}$ considers the vector $\vec{X}_k = (x_{k,1}, \ldots, x_{k,n})^{\top}$ and selects $\vec{b}_k \in \mathbb{Z}_p^n$ such that

$$\vec{b}_k{}^{\top} \cdot M_{\vec{X}_k} = \vec{b}_k{}^{\top} \cdot \begin{pmatrix} -\frac{x_{k,2}}{x_{k,1}} & -\frac{x_{k,3}}{x_{k,1}} & \cdots & -\frac{x_{k,n}}{x_{k,1}} \\ & I_{n-1} & & \end{pmatrix} = \vec{0}. \tag{8}$$

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

Due to (8), the matrix B is defined in such a way that, for each $k \in [1, q]$, the k^{th} column of $M_{\vec{X}_k}^\top \cdot B \in (\mathbb{Z}_p)^{(n-1) \times n}$ is $\vec{0}$, so that $M_{\vec{X}_k}^\top \cdot B \cdot \vec{a}$ does not contain $a_k = \gamma^{n+1-k}$. Then, a private key for the identity ID_k (and thus the vector $\vec{X}_k$) can be obtained by implicity defining $\tilde{r}_k = r_k - \delta_0 \gamma^k$ for a random $r_k \stackrel{R}{\leftarrow} \mathbb{Z}_p$. Indeed, with the above choice of B, the first coordinate of $\vec{\alpha} = \vec{\delta} + \sum_{j=1}^q a_j \vec{b}_j$ 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

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

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

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 $\vec{Y} = (y_1, \ldots, y_n)^\top$ that must satisfy $\langle \vec{X}_k, \vec{Y} \rangle = 0$ for $k = 1$ to q. This amounts to say that $\vec{Y} = M_{\vec{X}_k} \cdot \vec{w}$, where $\vec{w} = (y_2, \ldots, y_n)^\top$ and for each $k \in [1, q]$, as we have the equivalence

$$\langle \vec{X}_k, \vec{Y} \rangle = 0 \quad \Leftrightarrow \quad y_1 = y_2 \cdot (-\frac{x_{k,2}}{x_{k,1}}) + \cdots + y_n \cdot (-\frac{x_{k,n}}{x_{k,1}}) \quad \Leftrightarrow \quad \vec{Y} = M_{\vec{X}_k} \cdot (y_2, \ldots, y_n)^\top.$$

Now, we claim that $\vec{Y}^\top \cdot B \cdot \vec{a} = 0$. Indeed,

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

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

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

for a random bit $\sigma \stackrel{R}{\leftarrow} \{0, 1\}$. 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 $\sigma \in \{0, 1\}$ 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 KP-ABE (where the reduction will set up part of the public parameters in a similar way to the proof of theorem 3) in Section 5.2 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 [5, 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 [5] or in LSW2 [32]) but provides significantly shorter ciphertexts (as the ciphertext overhead is decreased by more than 75%)[8] 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 [5, Sect.5.1]. Still, the new scheme is slightly more efficient and relies on the q-DBDHE assumption which is somewhat more natural than the stronger q-type assumption (MEBDH) used in [32, 5].

In comparison with the schemes of Lewko, Sahai and Waters [32], 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.

Table 1. Performances of revocation systems

Schemes	Ciphertext overhead $\lvert\mathbb{G}\rvert$	Private key size $\lvert\mathbb{G}\rvert$	Decryption cost pair.	exp.	Assumption
LSW1 [32]	$(2\bar{n}+1)$	3	3	$O(\bar{n})$	n-MEBDH
LSW2 [32]	$(2\bar{n}+7)$	7	9	$O(\bar{n})$	DLIN, DBDH
AL1 [5]	3	$(n+2)$	3	$O(n)$	n-MEBDH
AL2 [5]	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.

5.2 A Non-Monotonic KP-ABE Scheme with Short Ciphertexts

Ostrovsky, Sahai and Waters [36] 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 [36], the above technique was combined with the Naor-Pinkas revocation method [35] to cope with non-monotonic access structures. Lewko, Sahai and Waters provided improvements using a revocation system with short keys [32] instead of [35]. 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 4.3 in order to handle non-negated attributes.

[8] 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 [22].

▶ $\mathsf{Setup}(\lambda, n)$: 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 $\vec{H} = (h_1, \ldots, h_n)^\top$ and $\vec{U} = (u_0, \ldots, u_n)^\top$ 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 $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \xleftarrow{R} \mathbb{Z}_p^n$ and $\vec{\beta} = (\beta_0, \beta_1, \ldots, \beta_n)^\top \xleftarrow{R} \mathbb{Z}_p^{n+1}$. 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

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

▶ $\mathsf{Keygen}(\mathsf{msk}, \tilde{\mathbb{A}})$: 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 $\vec{\rho}_i = (\rho_{i,1}, \ldots, \rho_{i,n})^\top = (1, x_i, x_i^2, \ldots, x_i^{n-1})^\top$. 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 $D_i = (D_{i,1}^{(1)}, D_{i,2}^{(2)}, K_{\vec{\rho}_i,i}^{(3)}) \in \mathbb{G}^{n+1}$ 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

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

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

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

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 $\vec{Y} = (y_1, \ldots, y_n)^\top$ 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 $q+2 \le j \le n$. Then, it randomly picks $s \xleftarrow{R} \mathbb{Z}_p$ 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 $\omega \notin \tilde{\mathbb{A}}$. Otherwise, since $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some access structure $\mathbb{A}$ associated with a linear secret sharing scheme Π, 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 $\vec{Y} = (y_1, \ldots, y_n)^\top$ 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 $x_i \notin \omega$), the receiver sets $\vec{\rho}_i = (1, x_i, \ldots, x_i^{n-1})^{\top}$ and successively computes

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

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

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

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

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

so that only three pairing evaluations are necessary.

Theorem 4. *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. Concretely, any selective-set adversary $\mathcal{A}$ implies a n-DBDHE distinguisher $\mathcal{B}$ with advantage*

$$\mathbf{Adv}_{\mathcal{B}}^{n\text{-DBDHE}}(\lambda) \geq \mathbf{Adv}_{\mathcal{A}}^{\text{KP-ABE-sCPA}}(\lambda).$$

Proof. We outline 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 selective-set adversary $\mathcal{A}$. We define $\vec{\gamma} = (\gamma, \gamma^2, \ldots, \gamma^n)^{\top}$ for further use.

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

Simulation of the setup phase. To generate the master public key, $\mathcal{B}$ will consider three parts: the first part relates to non-negated attributes, which are elements $\vec{U} = g^{\vec{\beta}}$; the second part relates to negated attributes, which are elements $\vec{H} = g^{\vec{\alpha}}$; the last part is the common element $e(g,g)^{\alpha}$.

- For the common part, it picks $\delta_0 \xleftarrow{R} \mathbb{Z}_p$ 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 4.3 (which we omitted the proof there). More concretely, it picks $\theta_0 \xleftarrow{R} \mathbb{Z}_p$ and computes $u_0 = g^{\theta_0} \cdot g^{-\langle \vec{\gamma}, \vec{Y} \rangle}$ from $g^{\vec{\gamma}}$. Other components of $\vec{U}$ are defined by setting $\vec{U}' := (u_1, \ldots, u_n)^{\top} = g^{\vec{\gamma}} \cdot g^{\vec{\theta}}$, for some randomly chosen vector $\vec{\theta} \xleftarrow{R} \mathbb{Z}_p^n$, so that we have $\vec{\beta}' := (\beta_1, \ldots, \beta_n)^{\top} = \vec{\gamma} + \vec{\theta}$.

- For the public key part related to negated attributes, it simulates similarly as in the proof of the underlying IBR of Section 5.1 (which is recorded in the proof of theorem 3). Intutively, it proceeds as if the announced set $\tilde{S}$ in theorem 3 (for private key queries there) is set to $\tilde{S} = \omega^\star$. More concretely, we first write $\omega^\star = \{\omega_1, \ldots, \omega_q\}$ in some order, then we define their corresponding vectors $\vec{X}_1, \ldots, \vec{X}_q$ as $\vec{X}_k = (1, \omega_k, \ldots, \omega_k^{n-1})^\top$. It then defines the $n \times n$ matrix $B = \left(\vec{b}_1 | \ldots | \vec{b}_q | \vec{0} | \ldots | \vec{0}\right)$ from the definition of $\vec{b}_k$ as in equation (8), where it can be re-written this time as:

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

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

Simulation of key extraction queries. 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 Π. Let $L \in \mathbb{Z}_p^{\ell \times n}$ denote the share-generating matrix for Π. 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, $\vec{1} = (1, 0, \ldots, 0)^\top$ 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 2, due to the proposition 11 in [32], we have that there must exist an efficiently computable vector $\vec{w} \in \mathbb{Z}_p^n$ such that $\langle \vec{1}, \vec{w} \rangle = 1$ and $L_{\omega'} \cdot \vec{w} = \vec{0}$. Now $\mathcal{B}$ will implicitly define each share of α as $\lambda_i = \langle \vec{L}_i, \vec{v} \rangle$, 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 $\vec{v} = \vec{\zeta} + (\alpha - \zeta_1)\vec{w}$ where $\vec{\zeta} = (\zeta_1, \ldots, \zeta_n)^\top \xleftarrow{R} \mathbb{Z}_p^n$. Note that we have that $v_1 = \alpha$ and that $v_2, \ldots, v_n \in \mathbb{Z}_p$ are uniformly distributed, as required in Definition 4. Although $\mathcal{B}$ cannot compute $\langle \vec{L}_i, \vec{v} \rangle$ 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 $\breve{x}_i \notin \omega'$), $\lambda_i = \langle \vec{L}_i, \vec{v} \rangle$ 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

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

    where $\vec{\rho}_i = (\rho_{i,1}, \ldots, \rho_{i,n})^\top = \vec{X}_k = (1, \omega_k, \ldots, \omega_k^{n-1}) = (1, x_i, \ldots, x_i^{n-1})$, 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 $\breve{x}_i \in \omega'$), $\langle \vec{L}_i, \vec{w} \rangle = \vec{0}$ so that $\vec{L}_i \cdot \vec{v} = \vec{L}_i \cdot \vec{\zeta}$ 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 $D_{i,1}^{(2)} = g^{\vec{L_i}\cdot\vec{v}} \cdot h_1^{r_i}$ 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 $x_i \in \omega^\star$, $\lambda_i = \langle \vec{L_i}, \vec{v} \rangle$ 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 $x_i \notin \omega^\star$, $\lambda_i = \langle \vec{L_i}, \vec{v} \rangle$ 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 [13][Theorem 1]. Namely, it considers the $n \times (n-1)$ matrix

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

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

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

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

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

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

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

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

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

    is also computable since the coefficient of γ^{n+1} is $-\delta_0$ in the product $-\tilde{r}\langle \vec{\gamma}, \vec{Y} \rangle$. 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.

Challenge. To generate the challenge ciphertext, $\mathcal{B}$ proceeds almost exactly as in the proof of theorem 3. Due to the choice of $\vec{U}$ and $\vec{H}$ in the setup phase, we have $u_0 \cdot g^{\langle \vec{\beta}', \vec{Y} \rangle} = g^{\theta_0 + \langle \vec{\theta}, \vec{Y} \rangle}$ and $g^{\langle \vec{\alpha}, \vec{Y} \rangle} = g^{\langle \vec{\delta}, \vec{Y} \rangle}$, so that the simulator $\mathcal{B}$ can flip a random coin $\sigma \xleftarrow{R} \{0,1\}$ and calculate

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

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_σ whereas it perfectly hides the bit $\sigma \in \{0,1\}$ if $T \in_R \mathbb{G}_T$. □

5.3 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 [41] is slightly different from the original one [28]. Basically, in the traditional definition of KP-ABE, an attribute can be represented in arbitrary formats; while on the other hand, in the definition from [41], an attribute is required to be represented in the name-value (or label-value) pair format. For self-containment, we re-formalize the latter in Appendix A, where we also briefly propose a modification of KP-ABE [41] so as to have the same functionality as the original ABE. We also include this modified scheme in Table 2. Note that [41] has a unique feature of being adaptively secure.

Regarding the size of public keys, only the scheme of [41] has the size (n^2). In contrast, all the other schemes have the same linear dependency on n which disappears in the random oracle model (where these $O(n)$ public group elements can be derived by applying a random oracle to some short pre-determined strings).

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

Schemes	Ciphertext overhead $\lvert\mathbb{G}\rvert$	Private key size $\lvert\mathbb{G}\rvert$	Decryption cost pair.	exp.	Assumption
OSW [36]	$O(\bar{n})$	$O(t \cdot \log n)$	$O(t)$	$O(t \cdot \bar{n})$	DBDH
LSW [32]	$O(\bar{n})$	$O(t)$	$O(t)$	$O(t \cdot \bar{n})$	n-MEBDH
OT [41]	$O(\bar{n} \cdot \varphi)$	$O(t \cdot \varphi)$	$O(t \cdot \varphi)$	$O(t)$	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

† $\bar{n} = |\text{attribute set}| = |\omega|$ 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.

6 Concluding Remarks

This paper presented the first results for expressive ABE schemes with constant-size ciphertexts. In the future, several open questions deserve further investigations.

Shorter keys. First, it will be interesting to see if shorter private keys can be obtained in non-monotonic schemes without affecting their expressivity or the size of ciphertexts. A sufficient condition would be to obtain identity-based broadcast encryption and revocation schemes that satisfy our template and simultaneously provide short ciphertexts and sub-linear-size private keys.

Full Security. Another problem is to attain full security with compact ciphertexts. At first glance, the techniques of [33] may seem to apply to the monotonic KP-ABE implied by Section 4 since the latter bears some similarities with the large-universe schemes of [38, 28]: at a high level, all these schemes can be seen as relatives of the first selectively secure Boneh-Boyen IBE [8].

Unfortunately, we were not able to adapt the proof techniques of Lewko *et al.* [33] in our setting. These techniques make use of a sequence of games and a crucial step of the proof consists of a game transition where so-called semi-functional components are introduced in private keys. This transition is justified by an indistinguishability assumption and an information-theoretic argument

according to which, as long as the adversary does not make illegal private key queries, he will not be able to notice a correlation between the semi-functional components of ciphertexts and private keys. If we try to apply the same ideas to the scheme described in Section 4 (when the latter is instantiated in composite order groups as in [33]), the proof fails in the crucial step because of our longer private keys which prevent us from hiding the correlation between semi-functional ciphertext/key components in the information-theoretic sense.

Simpler Assumptions and More Expressive Ciphertext-Policy ABE. Another worthy goal is the realization of ABE with short ciphertexts under simple assumptions (*i.e.*, assumptions of constant-size such as the Decision Linear assumption [10] or the Decision Bilinear Diffie-Hellman assumption [11]) instead of "q-type" assumptions that were used in this paper.

In the ciphertext-policy setting, yet another challenging problem is to achieve the same level of expressivity and efficiency (or prove it is impossible) as in the key-policy case. Our intuition is that it will be difficult to do much better than the scheme of Section 3 in these regards. Handling complex boolean formulas would require to encode them within a constant number of group elements and we are not aware of a method to do this.

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, 2007.
2. S. Al-Riyami, J. Malone-Lee, N.-P. Smart. Escrow-free encryption supporting cryptographic workflow. International Journal of Information Security, vol. 5 (4), pp. 217–229 (2006)
3. N. Attrapadung, H. Imai. Dual-Policy Attribute Based Encryption. In *ACNS'09*, *LNCS* 5536, pp. 168–185, 2009.
4. N. Attrapadung, H. Imai. Conjunctive Broadcast and Attribute-Based Encryption. In *Pairing'09*, *LNCS* 5671, pp. 248–265, 2009.
5. 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. Full version available from `http://perso.uclouvain.be/benoit.libert/functional-full-version.pdf`
6. N. Attrapadung, B. Libert, E. De Panfieu. Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In *PKC'11*, *LNCS* 6571, pp. 90–108. Springer, 2011.
7. J. Bethencourt, A. Sahai, B. Waters. Ciphertext-Policy Attribute-Based Encryption. IEEE Symposium on Security and Privacy (S&P), pp. 321-334, 2007.
8. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238, 2004.
9. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical Identity-Based encryption with Constant Size Ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005.
10. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Crypto'04*, LNCS 3152, pages 41–55, 2004.
11. 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.
12. 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.
13. D. Boneh, M. Hamburg. Generalized Identity Based and Broadcast Encryption Schemes. In *Asiacrypt'08*, *LNCS* 5350, pp. 455–470, 2008.
14. D. Boneh, A. Sahai, B. Waters. Functional Encryption: Definitions and Challenges. In *TCC'11*, *LNCS* 6597, pp. 253–273, 2011.
15. X. Boyen. General *Ad Hoc* Encryption from Exponent Inversion IBE. In *Eurocrypt'07*, *LNCS* 4515, pp. 394–411, 2007.
16. R. Canetti, S. Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *Eurocrypt'99*, LNCS 1592, Springer, pp. 90–106, 1999.
17. R. Canetti, S. Halevi, J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *Eurocrypt'03*, *LNCS* 2656, pp. 254–271, 2003.
18. R. Canetti, S. Halevi, J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Eurocrypt'04*, *LNCS* 3027, pp. 207–222, 2004.

19. Z. Chai, Z. Cao, Y. Zhou. Efficient ID-based broadcast threshold decryption in ad hoc network. In *Proc. of IMSCCS'06*, Volume 2, IEEE Computer Society, pp. 148–154, 2006.
20. M. Chase. Multi-authority Attribute Based Encryption. In *TCC'07*, *LNCS* 4392, pp. 515–534, 2007
21. M. Chase, S. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *ACM-CCS'09*, pp. 121–130, 2009.
22. J.-H. Cheon. Security Analysis of the Strong Diffie-Hellman Problem. In *Eurocrypt'06*, *LNCS* 4004, pp. 1–11, 2006.
23. L. Cheung, C. Newport. Provably secure ciphertext policy ABE. In *ACM-CCS'07*, pp. 456–465, 2007.
24. V. Daza, J. Herranz, P. Morillo, C. Ràfols. CCA2-secure threshold broadcast encryption with shorter ciphertexts. In *ProvSec'07*, LNCS 4784, Springer, pp. 35–50, 2007.
25. C. Delerablée, P. Paillier, D. Pointcheval. Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys. In *Pairing'07*, LNCS 4575, Springer, pp. 39–59, 2007.
26. C. Delerablée and D. Pointcheval. Dynamic threshold public-key encryption. In *Crypto'08*, LNCS 5157, Springer, pp. 317–334, 2008.
27. 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.
28. 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.
29. V. Goyal, A. Jain, O. Pandey, A. Sahai. Bounded Ciphertext Policy Attribute Based Encryption. ICALP (2) 2008, *LNCS* 5126, pp. 579–591, 2008.
30. J. Herranz, F. Laguillaumie, C. Ràfols. Constant-Size Ciphertexts in Threshold Attribute-Based Encryption. In *PKC'10*, *LNCS* 6056, Springer, 2010.
31. J. Katz, A. Sahai, B. Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *Eurocrypt'08*, *LNCS* 4965, pp. 146-162, 2008.
32. A. Lewko, A. Sahai, B. Waters. Revocation Systems with Very Small Private Keys. In IEEE Symposium on Security and Privacy (S&P) 2010.
33. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Eurocrypt'10*, *LNCS* 6110, pp. 62–91, 2010.
34. M. Naor. On Cryptographic Assumptions and Challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109, 2003.
35. M. Naor, B. Pinkas. Efficient Trace and Revoke Schemes. In *Financial Cryptography 2000*, *LNCS* 1962, pp. 1-20, 2000.
36. R. Ostrovsky, A. Sahai, B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM-CCS'07*, pp. 195–203, 2007.
37. R. Sakai, M. Kasahara. ID-based Cryptosystems with Pairing on Elliptic Curve. Cryptology ePrint Archive: Report 2003/054, 2003.
38. A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt'05*, *LNCS* 3494, pp. 457–473, 2005.
39. A. Shamir. How to share a secret. In *Communications of the ACM*, vol. 22, pp. 612–613, 1979.
40. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Crypto'84*, *LNCS* 196, pp. 47–53, 1984.
41. 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.
42. B. Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC 2011*, *LNCS* 6571, pp. 53–70, 2011.
43. S. Yamada, N. Attrapadung, G. Hanaoka, N. Kunihiro. Generic Constructions for Chosen-Ciphertext Secure Attribute Based Encryption. In *PKC 2011*, *LNCS* 6571, pp. 71–89, 2011.

A Variant: KP-ABE with Labeling

We re-formalize the KP-ABE definition of [41] 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 [41] further restricts the maximum repetition allowed for labels in one policy, which we denote by φ in Table 2.

Definition 11 (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.,* [28, 36] 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 [13, 5]. 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 [41]. We construct by instantiating the general KP-FE scheme of [41] with $d = 1$, and with the inner product relation being instantiated to IBBE, similarly as we did in Section 4.3, and setting the bound $\varphi = n$.

B The Boneh-Hamburg Spatial Encryption and IBBE Schemes

We recall the concept of spatial encryption [13]. For a matrix $M \in \mathbb{Z}_p^{n \times d}$ and a vector $\vec{c} \in \mathbb{Z}_p^n$, one considers the affine space $\mathrm{Aff}(M, \vec{c}) = \{M\vec{w} + \vec{c} \mid \vec{w} \in \mathbb{Z}_p^d\}$. 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

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

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 [13], 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 [9].

► $\mathsf{Setup}(\lambda, n)$: 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 $g \xleftarrow{R} \mathbb{G}$. Choose $\alpha, \alpha_0 \xleftarrow{R} \mathbb{Z}_p$ and a vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \xleftarrow{R} \mathbb{Z}_p^n$ to compute $h_0 = g^{\alpha_0}$, $\vec{H} = g^{\vec{\alpha}}$ and $e(g,g)^\alpha$. The master public key is $\mathsf{mpk} = (g,\ e(g,g)^\alpha,\ h_0,\ \vec{H} = g^{\vec{\alpha}})$ while the master secret key is $\mathsf{msk} = (\vec{\alpha}, \alpha_0, \alpha)$.

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

$$K_V = (K_1, K_2, K_3) = \left(g^\alpha \cdot h_0^r \cdot g^{r\langle \vec{x}, \vec{\alpha} \rangle},\ g^r,\ g^{rM^\top \vec{\alpha}}\right)$$

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

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

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

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

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

▶ $\mathsf{Decrypt}(\mathsf{mpk}, V, K_V, \vec{x}, C)$: 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 $g \xleftarrow{R} \mathbb{G}$. Choose $h_0 \xleftarrow{R} \mathbb{G}$ as well a vector $\vec{h} = (h_1, \ldots, h_n)^\top \xleftarrow{R} \mathbb{G}^n$ such that $h_i = g^{a_i}$ for $i = 1, \ldots, n$ with $\vec{a} = (a_1, \ldots, a_n)^\top \xleftarrow{R} \mathbb{Z}_p^n$. Finally, pick $\alpha \xleftarrow{R} \mathbb{Z}_p^*$ and compute $e(g,g)^\alpha$. The master public key is $\mathsf{mpk} = \big(g,\ e(g,g)^\alpha,\ h_0,\ \vec{h} = g^{\vec{a}}\big)$. while the master secret key is $\mathsf{msk} = (\vec{a}, \alpha)$.

▶ $\mathsf{Keygen}(\mathsf{msk}, \mathsf{ID})$: to generate a private key for an identity ID, choose $r \xleftarrow{R} \mathbb{Z}_p^*$ 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 $g^{r \cdot M_{\mathsf{ID}}^\top \cdot \vec{a}}$, 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 $s \xleftarrow{R} \mathbb{Z}_p^*$ 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

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

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

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

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

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

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

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

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

$$e(D_{\mathsf{ID}}, g) = e(g, g)^{\alpha} \cdot e(h_0 \cdot h_1^{\rho_1} \cdots h_{q+1}^{\rho_{q+1}}, d_{\mathsf{ID}}) \tag{9}$$

By raising both members of (9) 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 [13] 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 4.3 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 [5], a vector $\vec{X} = (x_1, \ldots, x_n)^{\top}$ of key attributes can be mapped onto a $(n-1)$-dimension affine space $V_{\vec{X}} = \mathrm{Aff}(M_{\vec{X}}, \vec{0}_n) = \{M_{\vec{X}}\vec{w} + \vec{0}_n \mid \vec{w} \in \mathbb{Z}_p^{n-1}\}$ with the matrix $M_{\vec{X}} \in \mathbb{Z}_p^{n\times(n-1)}$

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

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

$$\langle \vec{X}, \vec{Y} \rangle = 0 \Leftrightarrow y_1 = y_2 \cdot (-\frac{x_2}{x_1}) + \cdots + y_n \cdot (-\frac{x_n}{x_1}) \tag{10}$$

$$\Leftrightarrow \vec{Y} = M_{\vec{X}} \cdot (y_2, \ldots, y_n)^{\top} \ \Leftrightarrow\ \vec{Y} \in V_{\vec{X}}. \tag{11}$$