Multi-Authority ABE, Revisited.

Miguel Ambrona[1] and Romain Gay[2]

[1] Nomadic Labs
[2] IBM Research Zurich

Abstract. Attribute-Based Encryption (ABE) is a cryptographic primitive which supports fine-grained access control on encrypted data, making it an appealing building block for many applications. Multi-Authority Attribute-Based Encryption (MA-ABE) is a generalization of ABE where the central authority is distributed across several independent parties.
We provide the first MA-ABE scheme from prime-order pairings where no trusted setup is needed and where the attribute universe of each authority is unbounded. Our constructions rely on a common modular blueprint that uses an Identity-Based Functional Encryption scheme for inner products (ID-IPFE) as an underlying primitive. Our presentation leads to simple proofs of security and brings new insight into the algebraic design choices that seem common to existing schemes. In particular, the well-known MA-ABE construction by Lewko and Waters (EUROCRYPT 2011) can be seen as a specific instantiation of our modular construction.
Our schemes enjoy all of their advantageous features, and the improvements mentioned. Furthermore, different instantiations of the core ID-IPFE primitive lead to various security/efficiency trade-offs: we propose an adaptively secure construction proven in the generic group model and a selectively secure one that relies on SXDH. As in previous work, we rely on a hash function (to generate matching randomness for the same user across different authorities while preserving collusion resistance) that is modeled as a random oracle.

1 Introduction

Attribute-Based Encryption (ABE) [SW05, GPSW06] subsumes traditional public-key encryption by providing fine-grained access to the encrypted data. Namely, each ciphertext is associated with an access policy, and each user receives a so-called user secret key according to certain credentials. If these credentials fulfill the policy access, then the user secret key can be used to successfully decrypt the ciphertext. Otherwise, the plaintext remains hidden. In fact, security should hold even in the presence of colluding users.

Despite generating significant interest in the research community, the notion of ABE suffers from several drawbacks. Indeed, the user secret keys are generated from a so-called master secret key, which can potentially decrypt any generated ciphertext. Consequently, user secret key generation must be performed by a trusted third party, who controls such a master secret key and that must be online every time that a user secret key is requested (not just when setting up the scheme). This third party is a single point of failure for the system and will be a target of choice for adversaries. Spreading the master secret key among different users to alleviate this bottleneck will increase the chance of key exposure. Moreover, the trusted third party can impersonate any user of their choice, acting as an escrow (see [Rog15] for further details on the key-escrow issue faced by ABE). Furthermore, in many scenarios, the access policy used to generate a ciphertext includes attributes coming from different organizations.

To mitigate these shortcomings, [Cha07] and later [MKE08] considered a variation of ABE where any party can become an authority by publishing some public key; these authorities, created on the fly, handle different attributes, and no coordination is required among them. In fact, a user equipped with a global identifier can collect different credentials associated with different attributes from each authority. However, the user must then interact with a trusted central authority that will provide the ABE user secret keys.

[1] Most of this work was done while the author was employed by NTT Laboratories.

The advantage of their approach is that this central authority is agnostic to the meaning of the attributes and credentials of the user, and does not need to communicate with the actual authorities. However, most of the aforementioned shortcomings remain. Afterward, [LCLS08] removed the need for a central authority, but the set of authorities in their construction is fixed and they must interact during the setup phase. Another limitation is that the security of their scheme is only proven for an a priori bounded number of collusions. [CC09] also presented a scheme with no central authority relying on distributed PRF. However, their scheme is still limited in terms of expressiveness (it can only express a strict AND policy) and only handles a pre-determined set of attributes. In [LW11], the authors gave the first construction where there is no central authority, authorities can join the system on the fly without communicating with each other and the ciphertexts can be associated with a rich class of expressive access policies (including Boolean formula). Despite these impressive features, their construction still suffers from some limitations: it requires a trusted setup; it uses inefficient composite-order pairings; each authority can only handle a small (poly-size) set of attributes, in fact the public key of each authority grows with the number of attributes owned by the authority.

Our contribution. We address the disadvantages of [LW11]. Namely, we provide the first Multi-Authority ABE (MA-ABE) from prime-order pairings, where there is no trusted setup beyond the mere agreement of which groups and which hash function to use. Moreover, the attribute set of each authority is unbounded. Our constructions also keep the advantageous features of [LW11], in particular, the fact authorities can join the system dynamically and autonomously i.e without any interaction with other authorities. Individual authorities cannot generate a full-fledged user secret key, which means they cannot decrypt a ciphertext on their own, or impersonate a user. Only if sufficiently many authorities collude, should they be allowed to generate a given user secret key (that is the correctness of the scheme).

The use of prime-order pairing vastly improves the performance of our scheme (see [Gui13] for a comparison between composite vs prime-order pairings). Moreover, existing techniques to convert composite-order pairing to their prime-order counterpart (see e.g. [OT09, Fre10, Lew12, CGW15, Att16, AC16]) do not seem to a apply straightforwardly to the case of multi-authority ABE. However, we do not consider this achievement to be the most technically challenging part of our contribution. Instead, our most valuable technical insight is to remove some of the idiosyncrasies of Lewko-Waters' scheme that are detrimental for its efficiency. As an example, the construction by Lewko and Waters uses a target group element (whose order of magnitude is much larger than source group elements) in the ciphertext for no obvious reason. We instead adopt a more systematic approach, where all of our constructions rely on a common blueprint that makes use of practical Functional Encryption for simple function, namely, identity-based products (we refer to our technical overview for more details about FE). Apart from the practical features highlighted above, our principled viewpoint has the advantage that, by using different FE schemes, we can obtain diverse MA-ABE schemes with various security/efficiency trade-offs.

Our first construction uses an FE which is proven secure in the Generic Group Model (GGM). Note that we do not argue that the GGM is a good replacement of security reductions from standard assumptions. However, our result shows that using the GGM opens up to new qualitative improvements upon existing constructions (e.g. the unbounded attribute space). Moreover, recent standardization efforts (e.g., by the European Telecommunications Standards Institute, ETSI [Ins18]) call for truly practical schemes at the price of arguably aggressive security assumptions (note that the state of the art MA-ABE from [LW11] is patented; we plan to provide an open source implementation of our MA-ABE in the future). Our second construction uses an FE whose security is proven from standard pairing assumptions (namely, SXDH). The resulting MA-ABE is restricted in the sense that the access policies in the ciphertexts must use at most B attributes owned from a given authority, where B is an a priori bound. The total number of attributes used in the access policy, the number of authorities involved, or the attribute spaces of each authority are still unbounded (and in that sense, the scheme is still an improvement upon Lewko-Waters). The features are summarized in Table 1.

Finally, we prove that in our construction, the randomness used in the different FE ciphertexts contained in an MA-ABE ciphertext can be re-used (the MA-ABE encryption uses as sub-routine several calls to the FE encryption), at the price of more sophisticated security analysis. Doing so yields an improved decryption

Reference	[LW11]	scheme #1	scheme #2
$\lvert\mathsf{usk}\rvert$	$\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}_2\rvert$	$4B\lvert\mathbb{G}_2\rvert$
$\lvert\mathsf{ct}\rvert$	$3\ell(\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_\mathsf{t}\rvert)$	$(3\ell+t)\lvert\mathbb{G}_1\rvert$	$(2+4\ell)\lvert\mathbb{G}_1\rvert$
T_{Dec} (pairings)	2ℓ	$2+\ell+t$	$2(1+\ell+B)$
T_{Dec} (exponentiations)	ℓE_t	$\ell(2E_1+E_2)$	$2\ell(E_1+E_2)$
assumption	composite	GGM	SXDH
attribute universe	poly-size	unbounded	unbounded
attributes per authority	bounded	unbounded	bounded

Table 1. Comparison among MA-ABE schemes with respect to an access policy $(\boldsymbol{M}, \rho)$ where $\boldsymbol{M}$ has dimension $n \times \ell$, and $\rho : [\ell] \to \mathcal{U}$. E_s stand for exponentiation in $\mathbb{G}_s$, where $s \in \{1, 2, \mathsf{t}\}$, respectively. The group operations are omitted, since much more efficient than exponentiations and pairing operations. Note that [LW11] uses symmetric, composite-order groups, whereas we use asymmetric, prime-order groups, which are more efficient. The key encapsulation is omitted in the ciphertexts size. Here, t denotes the maximal number of attributes used in $(\boldsymbol{M}, \rho)$ owned by a given authority, i.e. $\max_{j\in[\ell]} \xi(j)$. When t is a priori bounded, we denote by B the bound. Note that in [LW11] this number is bounded by the size of the attribute universe of the authorities. Scheme #1 refers to the construction from Fig. 3, when instantiated with the ID-IPFE from Fig. 4. Scheme #2 refers to the construction from Fig. 5, when instantiated with the ID-IPFE from Fig. 2

procedure, with a first step that only computes exponentiations in source groups, and the second step that computes more costly pairing operations. Only the first step will involve a number of operations that grows with the size of the access policy in the ciphertext; the second step only incurs a number of operations that grows with the number of attributes owned by the same authority (this is typically much lower than the size of the policy itself, e.g. bounded by B in the second construction). Dealing with randomness re-use for optimized decryption, and handling unbounded attribute universe are the most significant technical challenges we face compared to prior works.

Technical overview. We consider an MA-ABE where access policies are represented by monotone span programs (MSP), as per Definition 1), which capture Boolean formulas. In a nutshell, a MSP allows users to produce shares $s_1, \ldots, s_\ell$ of a secret s, where ℓ is the size of the MSP, and each share s_j is associated with an attribute $\rho(j)$. The ABE uses cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{t}$ of prime order p, equipped with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathsf{t}$. We use additive bracket notations, where for all groups $s \in \{1, 2, \mathsf{t}\}$, all exponents $x \in \mathbb{Z}_p$, we write $[\![x]\!]_s = xP_s$ where P_s is a generator of $\mathbb{G}_s$.

For encryption, an exponent s is uniformly sampled from $\mathbb{Z}_p$ and the encapsulation key is defined as $[\![s]\!]_\mathsf{t}$ (we consider the KEM variant of ABE). The MSP is used to create shares $\{s_j\}_{j\in[\ell]}$ of s and shares $\{u_j\}_{j\in[\ell]}$ of 0. The MA-ABE ciphertext consists of one Functional Encryption (FE) of vector (s_j, u_j) per $j \in [\ell]$. The public key of the FE used for $j \in [\ell]$ is published by the authority that owns the attribute $\rho(j)$. That is, to register into the system, each authority runs the FE setup algorithm to create its pair of keys $(\mathsf{FE.pk}, \mathsf{FE.msk})$.

The FE we are using is for identity-based inner products. That is, each ciphertext encrypts a vector $\boldsymbol{x}$ (of some fixed dimension, say d, which is then set to 2 for our modular construction), and an identity id. Each functional secret key is associated with a vector $[\![\boldsymbol{y}]\!]_2 \in \mathbb{G}_2^d$ and an identity id'. The decryption of the ciphertext with the functional secret key succeeds if the identities match, in which case it recovers the inner product $[\![\boldsymbol{x}^\top \boldsymbol{y}]\!]_\mathsf{t}$. Nothing else is revealed about the encrypted vector $\boldsymbol{x}$. However, we do not require that the identities id and id' or the vector $[\![\boldsymbol{y}]\!]_2$ remain hidden. These functional secret keys can be generated from the master secret key of the FE scheme.

The MA-ABE ciphertext will contain the FE encryption of vector (s_j, u_j) with the identity set to be the attribute $\rho(j)$, under the $\mathsf{FE.pk}$ of the authority that owns the attribute $\rho(j)$, for all $j \in [\ell]$.

The secret key of a user identified by a global identifier gid, for an attribute a, will contain the FE functional secret key for vector $[\![1, z_{\mathsf{gid}}]\!]_2$ and identity a, where $[\![z_{\mathsf{gid}}]\!]_2$ is the output of the hash value $H(\mathsf{gid})$. This FE functional secret key is computed using the FE master secret key of the authority that owns the attribute a.

The user gid collects all the FE functional secret keys $\mathsf{sk}_{[\![1,z_{\mathsf{gid}}]\!]_2,\mathsf{a}}$, by making a request $(\mathsf{a}, \mathsf{gid})$ to the relevant authorities. Each FE key $\mathsf{sk}_{[\![1,z_{\mathsf{gid}}]\!]_2,\mathsf{a}}$ yields the value $[\![s_j + z_{\mathsf{gid}} u_j]\!]_{\mathsf{t}}$ for $\rho(j) = \mathsf{a}$. If sufficiently many such values are revealed, then they can be combined to obtain $[\![s + z_{\mathsf{gid}} \cdot 0]\!]_{\mathsf{t}} = [\![s]\!]_{\mathsf{t}}$, the encapsulation key. Here we rely on the fact that the share reconstruction for an MSP is linear. Otherwise said, if the user gid possesses enough attributes to satisfy the MSP in the ciphertext, it recovers the encapsulation key.

To argue security, we rely on the simulation security of the underlying FE scheme, which states that only $[\![s_j + z_{\mathsf{gid}} u_j]\!]_{\mathsf{t}}$ is revealed by the ciphertext and the FE functional secret key for identity $\rho(j)$ and vector $[\![1, z_{\mathsf{gid}}]\!]_2$ (together with the value $H(\mathsf{gid})$, which is public). Note that the term $[\![z_{\mathsf{gid}} u_j]\!]_{\mathsf{t}}$ prevents collusions across different gid. If for any given gid there are not enough attributes to satisfy the access structure associated to the ciphertext, then there are not enough values $[\![s_j + z_{\mathsf{gid}} u_j]\!]_{\mathsf{t}}$ to recover $[\![s]\!]_{\mathsf{t}}$.

We implement this general approach with different concrete FE schemes. The notion of identity-based inner product FE was originally put forth in [DP19, TT18] where they defined an indistinguishability-based security notion. However, for our MA-ABE we need simulation security. [ACGU20] gave the first simulation secure ID-inner-product FE, but the simulation security is only proven when the adversary gets one challenge ciphertext. Recall that in the IND-based setting, one challenge ciphertext implies many challenge ciphertexts via a standard hybrid argument, but this is not the case in the simulation setting. In fact, an incompressibility argument shows that simulation secure ID-inner-product FE for B challenge ciphertexts must have parameters (either the ct or sk size) growing with B, in the standard model. Imposing a bound on the challenge ciphertexts implies a bound on the number of attributes owned by a given authority in our MA-ABE. This is essentially our construction from standard assumptions. Another limitation is that our ID-inner-product FE is only selectively secure. On the plus side, we show that security also holds when the same randomness is used for different FE instances. This way, instead of decrypting all FE ciphertexts one by one naively, one can leverage the simple structure of the FE ciphertexts to combine them, obtaining an aggregated ciphertext (this only involves computing exponentiations in source groups) that only then is paired with the functional secret keys (the pairing computation cost far exceeds the cost of computing exponentiations in the source groups). To circumvent these limitations, we also provide an ID-inner-product FE which is *adaptively* simulation secure for an *unbounded* number of ciphertexts, with randomness re-use, proven in the GGM.

All of these schemes (and also prior schemes) crucially rely on the pseudo-randomness of the values $[\![z_{\mathsf{gid}}]\!]_2$ generated by the hash function. Intuitively, the terms computed during decryption cancel out when using the same gid, but combining values $[\![z_{\mathsf{gid}}]\!]_2$ for different in an attempts to get extra information that provided by the correctness of the scheme will fail. This formally argued in the random oracle model.

Related works. [Kim19] builds a multi-authority ABE for all circuits from LWE for a slightly different notion that the GID model presented here (it can be seen as a relaxation of the GID model). In a very recent work, [DKW21] builds an MA-ABE for DNF formula from LWE. In [MJ18], the authors present a decentralized ABE, which is similar to an MA-ABE except the number of authorities of the system is fixed ahead of time, and each authority requires the public keys of the other authorities to generate its share of the user secret key. They realized this notion for the orthogonality-testing predicate, which captures NC_0 circuits. Later on, [AGT20] extended their construction to partially hide the predicate in the user secret keys. In the same paper, they also presented a distributed ciphertext-policy ABE for NC_1, based on the LWE assumption and the bilinear generic group model. A distributed ABE is like an MA-ABE except the number of authorities is fixed ahead of time, and the adversary cannot create corrupted authority with arbitrary public keys, but is instead restricted to (statistically) recovers the secret key of honestly generated authorities.

Open problems: building an MA-ABE with the desirable *qualitative* efficiency features offered by the GGM, namely unbounded attribute space, and access structure with an unbounded number of attributes owned by a given authority, from standard assumptions. A different paradigm that does not rely on simulation security of the FE scheme would likely be required. Achieving adaptive security from standard assumptions would also be worthwhile (although this would qualify more as a quantitative than a qualitative improvement, since a guessing argument already provides adaptive security with an exponential security loss). Also, we believe the GGM is a valuable tool that leaves a lot of room for creativity when designing pairing-based constructions; it deserves further investigation, for the sake of expanding the functionality of existing schemes.

2 Preliminaries

2.1 Notations

We say a function $f : \mathbb{N} \to \mathbb{R}$ is negligible if f is asymptotically dominated by the inverse of any polynomial, i.e for every polynomial $p \in \mathbb{R}[X]$, there exists $\lambda_p \in \mathbb{N}$ such that $|f(\lambda)| \leq |1/p(\lambda)|$ for all $\lambda \geq \lambda_p$. We denote by $|\boldsymbol{v}|$ the length or dimension of vector $\boldsymbol{v}$ and by v_i its i-th component. For any $n \in \mathbb{N}$, we denote $\{1, \ldots, n\}$ by $[n]$. For any column vector $\boldsymbol{u} \in \mathbb{Z}^n$ and $\boldsymbol{v} \in \mathbb{Z}^m$, we denote by $(\boldsymbol{v}, \boldsymbol{u}) \in \mathbb{Z}^{n+m}$ the column vector obtained by concatenating them. Given two matrices (or vectors) $\boldsymbol{A} \in \mathbb{Z}^{m_1 \times n_1}$ and $\boldsymbol{B} \in \mathbb{Z}^{m_2 \times n_2}$, we denote by $\boldsymbol{A} \otimes \boldsymbol{B} \in \mathbb{Z}^{m_1 m_2 \times n_1 n_2}$ their Kronecker product, aka. tensor product defined as follows. For all $i \in [m_1 m_2]$ and $j \in [n_1 n_2]$ which we can write $i = m_1 i_1 + i_2$ with $i_1 \in [m_2]$, $i_2 \in [m_2]$, $j = n_1 j_1 + j_2$ with $j_1 \in [n_2]$, $j_2 \in [n_2]$, the (i, j)'th coordinate of $\boldsymbol{A} \otimes \boldsymbol{B}$ is $a_{i_1, j_1} \cdot b_{i_2, j_2}$.

2.2 Access Structure

We recall the definition of monotone access structures using the language of monotone span programs [KW93], which capture Boolean formulas.

Definition 1 (access structure [Bei96, KW93]). *A* monotone access structure *for attribute universe $\mathcal{U}$ is a pair $(\boldsymbol{M}, \rho)$ where $\boldsymbol{M} \in \mathbb{Z}_p^{n \times \ell}$ and $\rho : [\ell] \to \mathcal{U}$. The matrix $\boldsymbol{M}$ is used to generate shares as described in Fig. 1, and ρ maps each share to its associated attribute. Given a set of attributes $\mathcal{S} \subseteq \mathcal{U}$, we say that*

$$\mathcal{S} \text{ satisfies } (\boldsymbol{M}, \rho) \text{ iff } \mathbf{1} \in \mathsf{Span}(\boldsymbol{M}_{\mathcal{S}}),$$

Here, $\mathbf{1} := (1, 0, \ldots, 0) \in \mathbb{Z}^n$; $\boldsymbol{M}_S$ denotes the collection of vectors $\{\boldsymbol{M}_j : \rho(j) \in \mathcal{S}\}$ where $\boldsymbol{M}_j$ denotes the j'th column of $\boldsymbol{M}$; and Span *refers to linear span of collection of vectors over $\mathbb{Z}_p$.*

That is, $\mathcal{S}$ satisfies $(\boldsymbol{M}, \rho)$ iff there exists constants $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_p$ such that

$$\textstyle\sum_{\rho(j) \in S} \omega_j \boldsymbol{M}_j = \mathbf{1} \tag{1}$$

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

$\mathsf{Share}(\boldsymbol{M} \in \mathbb{Z}_p^{n \times \ell}, \alpha \in \mathbb{Z}_p)$:

Sample $\boldsymbol{u} \leftarrow_R \mathbb{Z}_p^{n-1}$, and for all $j \in [\ell]$, set $\alpha_j := (\alpha, \boldsymbol{u})^\top \boldsymbol{M}_j \in \mathbb{Z}_p$.

Return $\{\alpha_j\}_{j \in [\ell]}$.

Fig. 1. Share generation algorithm. Here, $\boldsymbol{M}_j$ denotes the j-th column of $\boldsymbol{M}$.

2.3 Pairing Groups

Let GGen be a PPT algorithm that on input the security parameter 1^λ, outputs a description $\mathcal{PG} = (p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathbb{G}_\mathsf{t}, e)$ of pairing groups where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_\mathsf{t}$ are cyclic groups of order p for a 2λ-bit prime p; P_1 and P_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathsf{t}$ is an efficiently computable (non-degenerate) bilinear map, thus $P_\mathsf{t} := e(P_1, P_2)$ generates $\mathbb{G}_\mathsf{t}$.

We use implicit representation of group elements. For $s \in \{1, 2, \mathsf{t}\}$ and $a \in \mathbb{Z}_p$, define $[\![a]\!]_s = a \cdot P_s \in \mathbb{G}_s$ as the implicit representation of a in $\mathbb{G}_s$. More generally, for a matrix $\boldsymbol{A} = (a_{ij}) \in \mathbb{Z}_p^{n \times m}$ we define $[\![\boldsymbol{A}]\!]_s$ as the implicit representation of $\boldsymbol{A}$ in $\mathbb{G}_s$:

$$[\![\boldsymbol{A}]\!]_s := \begin{pmatrix} a_{11} \cdot P_s \ \ldots \ a_{1m} \cdot P_s \\ \\ a_{n1} \cdot P_s \ \ldots \ a_{nm} \cdot P_s \end{pmatrix} \in \mathbb{G}_s^{n \times m}.$$

Given $[\![a]\!]_1$ and $[\![b]\!]_2$, one can efficiently compute $[\![a \cdot b]\!]_\mathsf{t}$ using the pairing e. For matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of matching dimensions, define $e([\![\boldsymbol{A}]\!]_1, [\![\boldsymbol{B}]\!]_2) := [\![\boldsymbol{AB}]\!]_\mathsf{t}$. For any matrix $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{Z}_p^{n \times m}$, any group $s \in \{1, 2, \mathsf{t}\}$, we denote by $[\![\boldsymbol{A}]\!]_s + [\![\boldsymbol{B}]\!]_s = [\![\boldsymbol{A} + \boldsymbol{B}]\!]_s$.

Definition 2 (DDH assumption). *For any adversary $\mathcal{A}$, any group $s \in \{1, 2, \mathsf{t}\}$ and any security parameter λ, let*

$$\mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_s, \mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathcal{PG}, [\![\boldsymbol{a}]\!]_s, [\![\boldsymbol{a}r]\!]_s)] - \Pr[1 \leftarrow \mathcal{A}(\mathcal{PG}, [\![\boldsymbol{a}]\!]_s, [\![\boldsymbol{u}]\!]_s)]|,$$

where the probabilities are taken over $\mathcal{PG} \leftarrow_R \mathsf{GGen}(1^\lambda, d)$, $\boldsymbol{a} \leftarrow_R \mathbb{Z}_p^2$, $r \leftarrow_R \mathbb{Z}_p$, $\boldsymbol{u} \leftarrow_R \mathbb{Z}_p^2$, and the random coins of $\mathcal{A}$. We say DDH holds in $\mathbb{G}_s$ if for all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_s, \mathcal{A}}(\lambda)$ is a negligible function of λ.

Definition 3 (SXDH assumption). *For any security parameter λ and any pairing group $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2, e) \leftarrow_R \mathsf{GGen}(1^\lambda)$, we say SXDH holds in $\mathcal{PG}$ if DDH holds in $\mathbb{G}_1$ and $\mathbb{G}_2$.*

3 ID-Based Inner-Product Functional Encryption

First we recall the notion of identity-based inner-product functional encryption — note that a similar definition was already given in [DP19], although with a weaker security notion. We provide a simulation-based security definition in the multi-instance setting, and we define a so-called decomposable structural property which is satisfied by most existing schemes. Then we present a variant of the ID-based IPFE from [ACGU20], which we prove simulation secure in the multi-instance setting under the SXDH assumption (the original proof was given in the weaker single-instance setting).

3.1 Definition

Let d be a polynomial. An identity-based inner product functional encryption scheme (ID-IPFE) for d-dimensional vectors consists of the PPT algorithms described below. For our purposes, we consider a two-step setup where global parameters are generated first, then used to derive the public key and master secret key. Note that the global parameters can be re-used for generating independent public key, master secret key pairs. We restrict our attention to schemes where the global parameters contain the description of a pairing group.

- $\mathsf{GlobalSetup}(1^\lambda) \to \mathsf{gp}$. On input the security parameter λ in unary, it outputs the global parameters gp, which contain a pairing group $(p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathbb{G}_\mathsf{t}, e)$. The global parameters gp are (sometimes implicitly) given as input to all other algorithms described here.
- $\mathsf{Setup}(\mathsf{gp}) \to (\mathsf{msk}, \mathsf{pk})$. On input the global parameters gp, it outputs a master secret key msk and a public key pk, defining an identity space $\mathcal{I}$ and randomness space $\mathcal{R}$.

- $\mathsf{Enc}(\mathsf{pk}, \boldsymbol{x}, \mathsf{id}; r) \to \mathsf{ct}_{\mathsf{id}}$. On input the public key pk, a message $\boldsymbol{x} \in \mathbb{Z}_p^{d(\lambda)}$, an identity $\mathsf{id} \in \mathcal{I}$, and the random coins $r \leftarrow_R \mathcal{R}$, the encryption algorithm outputs a ciphertext $\mathsf{ct}_{\mathsf{id}}$.
- $\mathsf{KeyGen}(\mathsf{msk}, [\![\boldsymbol{y}]\!]_2, \mathsf{id}) \to \mathsf{sk}_{\mathsf{id}}$. On input the master secret key msk, a vector $[\![\boldsymbol{y}]\!]_2 \in \mathbb{G}_2^{d(\lambda)}$ and an identity $\mathsf{id} \in \mathcal{I}$, the key generation algorithm outputs a functional secret key $\mathsf{sk}_{\mathsf{id}}$.
- $\mathsf{Dec}(\mathsf{ct}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id}'}, [\![\boldsymbol{y}]\!]_2) \to [\![z]\!]_{\mathsf{t}} / \perp$. The decryption algorithm gets a ciphertext $\mathsf{ct}_{\mathsf{id}}$, a functional secret key $\mathsf{sk}_{\mathsf{id}'}$ and vector $[\![\boldsymbol{y}]\!]_2$ associated to it. It outputs a group element in $\mathbb{G}_{\mathsf{t}}$ if $\mathsf{id} = \mathsf{id}'$, $\perp$ otherwise.

Identities are not intended to be hidden and we assume that ciphertexts and keys implicitly contain the identity they have been created for. To ease the notations, we write $d(\lambda)$ as d when clear from context.

Correctness. For all $\lambda \in \mathbb{N}$, all gp in the support of $\mathsf{GlobalSetup}(1^\lambda)$, all $(\mathsf{msk}, \mathsf{pk})$ in the support of $\mathsf{Setup}(\mathsf{gp})$, all $\mathsf{id} \in \mathcal{I}$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_p^d$:

$$\Pr\left[\mathsf{Dec}\left(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \boldsymbol{x}, \mathsf{id}),\ \mathsf{KeyGen}(\mathsf{msk}, [\![\boldsymbol{y}]\!]_2, \mathsf{id}), [\![\boldsymbol{y}]\!]_2\right) = [\![\boldsymbol{x}^\top \boldsymbol{y}]\!]_{\mathsf{t}}\right] = 1 \ ,$$

where the probability is taken over the random coins of Enc and KeyGen.

Decomposability. We say an ID-based inner-product FE is decomposable if each ciphertext consists of two parts, one header hd that is independent of the message $\boldsymbol{x}$ and the identity id, and one payload pl that depends on those. We additionally require that the payload can be computed deterministically from the header with the master secret key. More precisely, we have a PPT algorithm $\mathsf{Enc}_{\mathsf{hd}}$ and a PPT deterministic algorithm $\mathsf{Enc}_{\mathsf{pl}}$ such that for all $\lambda \in \mathbb{N}$, all gp in the support of $\mathsf{GlobalSetup}(1^\lambda)$, all $(\mathsf{pk}, \mathsf{msk})$ in the support of $\mathsf{Setup}(\mathsf{gp})$, all vectors $\boldsymbol{x} \in \mathbb{Z}_p^d$, all identities $\mathsf{id} \in \mathcal{I}$, the following distributions are identical:

$$\{\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, [\![\boldsymbol{x}]\!]_1, \mathsf{id}) : \mathsf{ct}\},$$
$$\{\mathsf{hd} \leftarrow \mathsf{Enc}_{\mathsf{hd}}(\mathsf{pk}), \mathsf{pl} = \mathsf{Enc}_{\mathsf{pl}}(\mathsf{msk}, \mathsf{id}, [\![\boldsymbol{x}]\!]_1, \mathsf{hd}) : (\mathsf{hd}, \mathsf{pl})\}.$$

Multi-instance simulation security. An ID-IPFE scheme is said to be multi-instance simulation secure if there exist a tuple of PPT algorithms $\mathsf{Sim} = (\overline{\mathsf{GlobalSetup}}, \overline{\mathsf{Setup}}, \overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}})$ such that for every admissible PPT adversary $\mathcal{A}$, there exists and a negligible function ν such that, for all $\lambda \in \mathbb{N}$, $|\Pr[\mathsf{Real}(\mathcal{A}, \lambda) \to 1] - \Pr[\mathsf{Ideal}(\mathsf{Sim}, \mathcal{A}, \lambda) \to 1]| \leq \nu(\lambda)$, where the experiments $\mathsf{Real}(\mathcal{A}, \lambda)$ and $\mathsf{Ideal}(\mathsf{Sim}, \mathcal{A}, \lambda)$ are defined below. An adversary is said to be admissible if it queries $\mathcal{O}_{\mathsf{Enc}}$ at most once on a given instance.

$\underline{\mathsf{Real}(\mathcal{A}, \lambda)}$:
$\mathsf{gp} \leftarrow \mathsf{GlobalSetup}(1^\lambda)$
$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Enc}}(\cdot,\cdot,\cdot), \mathcal{O}_{\mathsf{KG}}(\cdot,\cdot,\cdot), \mathcal{O}_{\mathsf{create}}}(\mathsf{gp})$
Output α.

$\underline{\mathsf{Ideal}(\mathsf{Sim}, \mathcal{A}, \lambda)}$:
$(\overline{\mathsf{gp}}, \mathsf{td}) \leftarrow \overline{\mathsf{GlobalSetup}}(1^\lambda)$
$\alpha \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Enc}}(\cdot,\cdot,\cdot), \mathcal{O}_{\mathsf{KG}}(\cdot,\cdot,\cdot), \mathcal{O}_{\mathsf{create}}}(\overline{\mathsf{gp}})$
Output α.

In the real experiment:

- On its i'th query, the oracle $\mathcal{O}_{\mathsf{create}}$ generates $(\mathsf{pk}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}(\mathsf{gp})$ and adds the pair $(\mathsf{pk}_i, \mathsf{msk}_i)$ to a list $\mathcal{L}$, initially empty. We denote by $|\mathcal{L}|$ the length of the list $\mathcal{L}$. It returns pk_i.
- The oracle $\mathcal{O}_{\mathsf{Enc}}$ samples the random coins $r \leftarrow_R \mathcal{R}$ once at the beginning of the experiment. Then, for each query $(\boldsymbol{x}, \mathsf{id}, i)$ it receives, if $i > |\mathcal{L}|$, it outputs nothing. Otherwise, it outputs $\mathsf{Enc}(\mathsf{pk}_i, \boldsymbol{x}, \mathsf{id}; r)$, where pk_i is in $\mathcal{L}$. Note that the same randomness $r \in \mathcal{R}$ is used to compute all ciphertexts. Recall that admissible adversaries can query $\mathcal{O}_{\mathsf{Enc}}$ on a given instance i at most once.
- When queried on $([\![\boldsymbol{y}]\!]_2, \mathsf{id}, i)$, if $i > |\mathcal{L}|$, it outputs nothing. Otherwise, it outputs $\mathsf{KeyGen}(\mathsf{msk}_i, [\![\boldsymbol{y}]\!]_2, \mathsf{id})$, where msk_i is in $\mathcal{L}$.

In the ideal experiment:

- On its i'th query, the oracle $\mathcal{O}_{\mathsf{create}}$ generates $(\overline{\mathsf{pk}}_i, \overline{\mathsf{msk}}_i) \leftarrow \overline{\mathsf{Setup}}(\overline{\mathsf{gp}})$ and adds the pair $(\overline{\mathsf{pk}}_i, \overline{\mathsf{msk}}_i)$ to a list $\mathcal{L}$, initially empty. It returns $\overline{\mathsf{pk}}_i$.
- When queried on $(\boldsymbol{x}, \mathsf{id}, i)$, provided $i \leq |\mathcal{L}|$, $\mathcal{O}_{\mathsf{Enc}}$ adds $(\boldsymbol{x}, \mathsf{id})$ to $\mathcal{Q}_{\mathsf{CT}}$ and outputs $\overline{\mathsf{Enc}}(\mathsf{td}, \overline{\mathsf{pk}}_i, \mathsf{id}, i, \mathit{leakage})$, where $\overline{\mathsf{pk}}_i$ is in $\mathcal{L}$. The case $i > |\mathcal{L}|$ is handled as in the real experiment.
- When queried on $([\![\boldsymbol{y}]\!]_2, \mathsf{id}, i)$, $\mathcal{O}_{\mathsf{KG}}$ adds $([\![\boldsymbol{y}]\!]_2, \mathsf{id})$ to $\mathcal{Q}_{\mathsf{KG}}$ and outputs $\overline{\mathsf{KeyGen}}(\mathsf{td}, [\![\boldsymbol{y}]\!]_2, \mathsf{id}, i, \mathit{leakage})$.

Here, *leakage* is defined based on the current state of the sets $\mathcal{Q}_{\mathsf{CT}}$ and $\mathcal{Q}_{\mathsf{KG}}$ that are initially empty, as follows:

$$\mathit{leakage} := \{([\![\boldsymbol{x}^\top \boldsymbol{y}]\!]_2, \mathsf{id}) \mid (\boldsymbol{x}, \mathsf{id}) \in \mathcal{Q}_{\mathsf{CT}} \wedge ([\![\boldsymbol{y}]\!]_2, \mathsf{id}) \in \mathcal{Q}_{\mathsf{KG}}\} \quad .$$

Selective security. we say the scheme is *selectively simulation secure* when security only holds for restricted adversaries that make all their queries to $\mathcal{O}_{\mathsf{Enc}}$ before any query to $\mathcal{O}_{\mathsf{KG}}$.

Remark on the leakage. Observe that the above definition assumes that the simulator has access to the inner product $[\![\boldsymbol{x}^\top \boldsymbol{y}]\!]_2$ in $\mathbb{G}_2$. In contrast, correctness allows one to recover the inner product only in $\mathbb{G}_\mathsf{t}$. This seemingly inconsistent definition makes it easier to design IB-IPFE constructions and argue their security, and it does not hinder the security of primitives which use IB-IPFE as a building block.

3.2 ID-based Inner-Product FE from SXDH

We recall the ID-based inner-product from [ACGU20] for vectors of dimension $d \in \mathsf{poly}(\lambda)$ with identity space $\mathbb{Z}_p$ in Fig. 2. It was originally proven simulation in the single instance from SXDH. We adapt the proof to the (stronger) multi-instance setting with randomness re-use, as defined in Section 3.1.

$\underline{\mathsf{GlobalSetup}(1^\lambda)}$:
Samples $\mathcal{PG} = (p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathbb{G}_\mathsf{t}, e) \leftarrow \mathsf{GGen}(1^\lambda)$, $\boldsymbol{a}, \boldsymbol{b} \leftarrow_R \mathbb{Z}_p^2$.
Return $\mathsf{gp} = (\mathcal{PG}, [\![\boldsymbol{a}]\!]_1, [\![\boldsymbol{b}]\!]_2)$.

$\underline{\mathsf{Setup}(\mathsf{gp})}$:
Sample $\boldsymbol{U}_0, \boldsymbol{U}_1 \leftarrow_R \mathbb{Z}_p^{2\times 2}$, $\boldsymbol{V} \leftarrow_R \mathbb{Z}_p^{d\times 2}$.
Set $\mathsf{msk} := (\boldsymbol{V}, \boldsymbol{U}_0, \boldsymbol{U}_1)$, $\mathsf{pk} := ([\![\boldsymbol{V}\boldsymbol{a}]\!]_1, [\![\boldsymbol{U}_0\boldsymbol{a}]\!]_1, [\![\boldsymbol{U}_1\boldsymbol{a}]\!]_1)$ and return $(\mathsf{msk}, \mathsf{pk})$.

$\underline{\mathsf{Enc}(\mathsf{pk}, \boldsymbol{x}, \mathsf{id};\ r)}$:
Given pk, $\boldsymbol{x} \in \mathbb{Z}_p^d$, $\mathsf{id} \in \mathbb{Z}_p$, and random coins $r \in \mathbb{Z}_p$,
return $\mathsf{ct} := ([\![\boldsymbol{a}r]\!]_1, [\![(\boldsymbol{U}_0 + \mathsf{id}\boldsymbol{U}_1)\boldsymbol{a}r]\!]_1, [\![\boldsymbol{V}\boldsymbol{a}r + \boldsymbol{x}]\!]_1) \in \mathbb{G}_1^{2+2+d}$.

$\underline{\mathsf{KeyGen}(\mathsf{msk}, [\![\boldsymbol{y}]\!]_2 \in \mathbb{G}_2^d, \mathsf{id} \in \mathbb{Z}_p)}$:
Sample $s \leftarrow_R \mathbb{Z}_p$ and return $\mathsf{sk} = ([\![\boldsymbol{b}s]\!]_2, [\![\boldsymbol{V}^\top \boldsymbol{y} + (\boldsymbol{U}_0 + \mathsf{id}\boldsymbol{U}_1)^\top \boldsymbol{b}s]\!]_2) \in \mathbb{G}_2^{2+2}$.

$\underline{\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}, [\![\boldsymbol{y}]\!]_2)}$:
Parse $\mathsf{ct} = ([\![\boldsymbol{c}_1]\!]_1, [\![\boldsymbol{c}_2]\!]_1, [\![\boldsymbol{c}_3]\!]_1) \in \mathbb{G}_1^2 \times \mathbb{G}_1^2 \times \mathbb{G}_1^d$.
Parse $\mathsf{sk} = ([\![\boldsymbol{k}_1]\!]_2, [\![\boldsymbol{k}_2]\!]_2, [\![\boldsymbol{y}]\!]_2) \in \mathbb{G}_2^2 \times \mathbb{G}_2^2 \times \mathbb{G}_2^d$.
Compute $[\![d]\!]_\mathsf{t} = e([\![\boldsymbol{c}_3^\top]\!]_1, [\![\boldsymbol{y}]\!]_2) \cdot e([\![\boldsymbol{c}_2^\top]\!]_1, [\![\boldsymbol{k}_1]\!]_2) / e([\![\boldsymbol{c}_1^\top]\!]_1, [\![\boldsymbol{k}_2]\!]_2)$.

Fig. 2. ID-based inner-product FE for d-dimensional vectors and ID space $= \mathbb{Z}_p$. Its multi-instance selective simulation security is proven under the SXDH assumption.

Correctness. For all $\lambda \in \mathbb{N}$, all gp in the support of $\mathsf{GlobalSetup}(1^\lambda)$, all $(\mathsf{pk}, \mathsf{msk})$ in the support of $\mathsf{Setup}(\mathsf{gp})$, all $\boldsymbol{x} \in \mathbb{Z}_p^d$, $\mathsf{id} \in \mathbb{Z}_p$, writing $\mathsf{ct}_{\mathsf{id}} = (\llbracket \boldsymbol{c}_1 \rrbracket_1, \llbracket \boldsymbol{c}_2 \rrbracket_1, \llbracket \boldsymbol{c}_3 \rrbracket_1)$ where $\llbracket \boldsymbol{c}_1 \rrbracket_1 \in \mathbb{G}_1^2$, $\llbracket \boldsymbol{c}_2 \rrbracket_1 \in \mathbb{G}_1^2$, $\llbracket \boldsymbol{c}_3 \rrbracket_1 \in \mathbb{G}_1^d$, and $\mathsf{sk}_{\mathsf{id}} = (\llbracket \boldsymbol{k}_1 \rrbracket_2, \llbracket \boldsymbol{k}_2 \rrbracket_2)$ where $\llbracket \boldsymbol{k}_1 \rrbracket_2 \in \mathbb{G}_2^2$, $\llbracket \boldsymbol{k}_2 \rrbracket_2 \in \mathbb{G}_2^2$, we have:

$$\begin{aligned}
&\llbracket \boldsymbol{c}_3^\top \boldsymbol{y} + \boldsymbol{c}_2^\top \boldsymbol{k}_1 - \boldsymbol{c}_1^\top \boldsymbol{k}_2 \rrbracket_{\mathsf{t}} \\
&= \llbracket \boldsymbol{y}^\top \boldsymbol{V}\boldsymbol{a}r + \boldsymbol{y}^\top \boldsymbol{x} + (\boldsymbol{b}s)^\top (\boldsymbol{U}_0 + \mathsf{id}\boldsymbol{U}_1)\boldsymbol{a}r - (\boldsymbol{b}s)^\top (\boldsymbol{U}_0 + \mathsf{id}\boldsymbol{U}_1)\boldsymbol{a}r - \boldsymbol{y}^\top \boldsymbol{V}\boldsymbol{a}r \rrbracket_{\mathsf{t}} \\
&= \llbracket \boldsymbol{x}^\top \boldsymbol{y} \rrbracket_{\mathsf{t}}.
\end{aligned}$$

Decomposability. We define PPT algorithms $\mathsf{Enc}_{\mathsf{hd}}$ and $\mathsf{Enc}_{\mathsf{pl}}$ as follows. Given as input the public key $\mathsf{pk} = (\llbracket \boldsymbol{V}\boldsymbol{a} \rrbracket_1, \llbracket \boldsymbol{U}_0\boldsymbol{a} \rrbracket_1, \llbracket \boldsymbol{U}_1\boldsymbol{a} \rrbracket_1)$, $\mathsf{Enc}_{\mathsf{hd}}(\mathsf{pk})$ samples $r \leftarrow_R \mathbb{Z}_p$ and outputs the header $\llbracket \boldsymbol{a}r \rrbracket_1 \in \mathbb{G}_1^2$. Given as input $\mathsf{msk} = (\boldsymbol{V}, \boldsymbol{U}_0, \boldsymbol{U}_1)$, an identity $\mathsf{id} \in \mathbb{Z}_p$, a message $\boldsymbol{x} \in \mathbb{Z}_p^d$, and a header $\mathsf{hd} = \llbracket \boldsymbol{c}_1 \rrbracket_1 \in \mathbb{G}_1^2$, $\mathsf{Enc}_{\mathsf{pl}}(\mathsf{msk}, \mathsf{id}, \boldsymbol{x}, \mathsf{hd})$ outputs the payload $\mathsf{pl} = (\llbracket (\boldsymbol{U}_0 + \mathsf{id}\boldsymbol{U}_1)\boldsymbol{c}_1 \rrbracket_1, \llbracket \boldsymbol{V}\boldsymbol{c}_1 + \boldsymbol{x} \rrbracket_1) \in \mathbb{G}_1^{2\times d}$. Note that pl is of the form $(\llbracket (\boldsymbol{U}_0 + \mathsf{id}\boldsymbol{U}_1)\boldsymbol{a}r \rrbracket_1, \llbracket \boldsymbol{V}\boldsymbol{a}r + \boldsymbol{x} \rrbracket_1)$, which the same as the payload computed by Enc.

Theorem 1 (Security). *The scheme in Fig. 2 is selective multi-instance simulation secure, assuming SXDH.*

The proof of this theorem is given in the Appendix C.

4 Definition of Multi-Authority ABE

We recall the definition of multi-authority ABE from [LW11]. We assume every authority is identified by a public key. For every authority pk, we denote by $\mathcal{U}_{\mathsf{pk}}$ the associated attribute universe. Without loss of generality, we assume that attribute universes are disjoint for different authorities.

We consider access structures $(\boldsymbol{M}, \rho)$ where $\boldsymbol{M} \in \mathbb{Z}_p^{n\times\ell}$, and ρ maps each row $j \in [\ell]$ to an attribute in $\mathcal{U}_{\theta(j)}$, where θ maps a row $j \in [\ell]$ to the authority who owns the attribute $\rho(j)$. Because several attributes used in $\boldsymbol{M}$ can be owned by the same authority, it is convenient for us to define a map $\xi : [\ell] \to \mathbb{N}$ that maps each row $j \in [\ell]$ to an ordinal number such that $\rho(j)$ is the $\xi(j)$'th attribute used in the access structure owned by authority $\theta(j)$, when ordered arbitrarily. To keep notations simple, we assume the maps θ and ξ are implicitly part of the description of the access structure.

We consider the following restriction on the access structures in the multi-authority setting.

Bounded number of attributes per authority in the ciphertexts: There exists a bound B such that any authority owns at most B attributes used by the access structure, i.e. for all $j \in [\ell]$, $\xi(j) \leq B$.

Definition. A MA-ABE scheme consists of the following PPT algorithms:

- $\mathsf{GlobalSetup}(1^\lambda) \to \mathsf{gp}$. On input the security parameter, it outputs global parameters, which are input to all other algorithms (usually implicitly).
- $\mathsf{AuthSetup}(\mathsf{gp}) \to (\mathsf{pk}, \mathsf{sk})$. Each authority runs a setup procedure to generate its own pair of keys. The public key serves as a univocal identifier for the authority, which is associated with an attribute universe denoted by $\mathcal{U}_{\mathsf{pk}}$.
- $\mathsf{Enc}(\boldsymbol{M}, \rho, \Pi) \to (\mathsf{ct}, \kappa)$. On input an access structure $\boldsymbol{M} \in \mathbb{Z}_p^{n\times\ell}$, $\rho : [\ell] \to \{0,1\}^*$ and a set of authorities Π such that for all columns $j \in [\ell]$, we have $\theta(j) \in \Pi$, the encryption algorithm outputs a ciphertext ct and a symmetric encryption key $\kappa \in \mathcal{K}$. The ciphertext implicitly contains a description of the access structure $(\boldsymbol{M}, \rho)$.
- $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \mathsf{gid}, \mathsf{a}) \to \mathsf{sk}_{\mathsf{gid},\mathsf{a}}$. On input an authority's public key pk and the corresponding secret key sk, a global identifier gid and an attribute $\mathsf{a} \in \mathcal{U}_{\mathsf{pk}}$, the key generation algorithm outputs a user secret key $\mathsf{sk}_{\mathsf{gid},\mathsf{a}}$, which implicitly contains a description of gid and a.

- $\mathsf{Dec}(\mathsf{ct}, \{\mathsf{sk}_{\mathsf{gid},\mathsf{a}}\}_{\mathsf{a}\in\mathcal{S}}) \to \kappa/\perp$. On input a ciphertext ct and a set of user secret keys $\{\mathsf{sk}_{\mathsf{gid},\mathsf{a}}\}_{\mathsf{a}\in\mathcal{S}}$ created for the same global identifier, decryption deterministically outputs a symmetric key κ or $\perp$.

Correctness. For all $\lambda \in \mathbb{N}$, all gp in the support of $\mathsf{GlobalSetup}(1^\lambda)$, all $\nu \in \mathbb{N}$, all $(\mathsf{pk}_1, \mathsf{sk}_1), \cdots, (\mathsf{pk}_\nu, \mathsf{sk}_\nu)$ in the support of $\mathsf{Setup}(\mathsf{gp})$, all access structures $(\boldsymbol{M}, \rho)$ associated with the set of authorities $\Pi = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_\nu\}$, all pairs (ct, κ) in the support of $\mathsf{Enc}(\boldsymbol{M}, \rho, \Pi)$, all sets of attributes $\mathcal{S} \subset \cup_{\mathsf{pk}\in\Pi}\mathcal{U}_{\mathsf{pk}}$ that satisfy $(\boldsymbol{M}, \rho)$ and all global identifiers $\mathsf{gid} \in \{0,1\}^*$:

$$\Pr\left[\mathsf{Dec}(\mathsf{ct}, \{\mathsf{sk}_{\mathsf{gid},\mathsf{a}}\}_{\mathsf{a}\in\mathcal{S}}) = \kappa\right] = 1 \ ,$$

where the probability is taken over $\mathsf{sk}_{\mathsf{gid},\mathsf{a}} \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \mathsf{gid}, \mathsf{a})$ for all $\mathsf{a} \in \mathcal{S}$ and where $\mathsf{pk} \in \Pi$ is the authority who owns attribute a.

Adaptive security. Given a multi-authority ABE denoted by ABE, for any stateful adversary $\mathcal{A}$ and security parameter λ, we define the advantage function:

$$\mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}}(\lambda) \coloneqq \left| \Pr\left[\begin{array}{c} \mathsf{gp} \leftarrow \mathsf{GlobalSetup}(1^\lambda) \\ (\boldsymbol{M}, \rho, \Pi_{\mathsf{hon}}, \Pi_{\mathsf{corr}}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{create}}, \mathcal{O}_{\mathsf{corr}}(\cdot), \mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot,\cdot)}(\mathsf{gp}) \\ (\mathsf{ct}^\star, \kappa) \leftarrow \mathsf{Enc}(\boldsymbol{M}, \rho, \Pi) \qquad : \beta' = \beta \\ \beta \leftarrow_R \{0,1\};\ K_0 \coloneqq \kappa;\ K_1 \leftarrow_R \mathcal{K} \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{corr}}(\cdot), \mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot,\cdot)}(\mathsf{ct}^\star, K_\beta) \end{array} \right] - \frac{1}{2} \right| \ .$$

The oracles are defined as follows:

- $\mathcal{O}_{\mathsf{create}}$: runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{AuthSetup}(\mathsf{gp})$, adds pk to the sets of honest authorities denoted by $\mathcal{S}_{\mathsf{hon}}$ (initially empty) and returns pk.
- $\mathcal{O}_{\mathsf{corr}}(\mathsf{pk})$: if $\mathsf{pk} \in \mathcal{S}_{\mathsf{hon}}$, it returns the associated secret key sk and removes pk from $\mathcal{S}_{\mathsf{hon}}$.
- $\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{pk}, \mathsf{gid}, \mathsf{a})$: if $\mathsf{pk} \in \mathcal{S}_{\mathsf{hon}}$, $\mathsf{a} \in \mathcal{U}_{\mathsf{pk}}$, it returns $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \mathsf{gid}, \mathsf{a})$, otherwise, it returns $\perp$.

The adversary $\mathcal{A}$ outputs an access structure $(\boldsymbol{M}, \rho)$ with respect to the authorities $\Pi_{\mathsf{hon}} \cup \Pi_{\mathsf{corr}}$, where Π_{hon} denotes the set of honest authorities, that is, which have been created via $\mathcal{O}_{\mathsf{create}}$, and which have not been queried to $\mathcal{O}_{\mathsf{corr}}$ (they can still be queried to $\mathcal{O}_{\mathsf{corr}}$ later on), whereas Π_{corr} denotes the set of corrupted authorities, that is, authorities created via $\mathcal{O}_{\mathsf{create}}$ that have been subsequently queried to $\mathcal{O}_{\mathsf{corr}}$, or authorities whose public key was maliciously created by the adversary $\mathcal{A}$ himself.

We require that Π_{corr} contains not only the public keys of the corrupted authorities, but also their associated secret keys[3].

We denote by $\mathcal{Q}_{\mathsf{KeyGen}}$ the set of queries to $\mathcal{O}_{\mathsf{KeyGen}}$, $\mathcal{S}_{\mathsf{hon}} \subseteq \Pi_{\mathsf{hon}}$ the set of authorities in Π_{hon} that are still honest at the end of the experiment, $\mathcal{S}_{\mathsf{corr}} = \Pi_{\mathsf{corr}} \cup \Pi_{\mathsf{hon}} \setminus \mathcal{S}_{\mathsf{hon}}$ and for all $\mathsf{gid} \in \{0,1\}^*$, $\mathcal{S}_{\mathsf{gid}} = \cup_{\mathsf{pk}\in\mathcal{S}_{\mathsf{corr}}}\mathcal{U}_{\mathsf{pk}} \cup \{\mathsf{a} \text{ s.t. } \exists \mathsf{pk} \in \mathcal{S}_{\mathsf{hon}}, (\mathsf{pk}, \mathsf{gid}, \mathsf{a}) \in \mathcal{Q}_{\mathsf{KeyGen}}\}$. We say the adversary $\mathcal{A}$ is admissible if for all $\mathsf{gid} \in \{0,1\}^*$, $\mathcal{S}_{\mathsf{gid}}$ does not satisfy $(\boldsymbol{M}, \rho)$ (as per Definition 1). We say ABE is adaptively secure if for all PPT admissible adversaries $\mathcal{A}$, there exists a negligible function ν such that for all $\lambda \in \mathbb{N}$, $\mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}}(\lambda) \leq \nu(\lambda)$.

Static corruptions. We say an ABE is secure with static corruptions if the adversary does not have access to the oracle $\mathcal{O}_{\mathsf{corr}}$. He can still create authorities maliciously as part of Π_{corr}, but all authorities created by $\mathcal{O}_{\mathsf{create}}$ remain honest throughout the experiment. Note that security with static corruptions implies adaptive corruptions via a standard guessing argument that incurs a security loss of 2^q where q denotes the number of queries to $\mathcal{O}_{\mathsf{create}}$.

[3] Note that this restriction of having to provide the secret keys of the corrupted authorities in Π_{corr} can be lifted via a generic use of Zero-Knowledge Argument of Knowledge. See Remark 1 for further details.

Selective security. We say an ABE is selectively secure if the adversary does not make any query to $\mathcal{O}_{\mathsf{KeyGen}}$ before receiving the challenge ciphertext $\mathsf{ct}^\star$. Similarly, selective security implies adaptive security via a standard guessing argument that incurs an exponential security loss.

Remark 1 (Stronger security via ZK-AoK). In the security definition above, we require the adversary to provide not only the public keys, but also the secret keys of all the authorities in Π_{corr}. It is possible to lift this restriction, and thereby strengthen the security definition, using standard techniques involving Zero-Knowledge Argument of Knowledge (ZK-AoK, see Definition 4). Any authority must publish not only a public key, but also an argument of knowledge of the associated secret key. The zero-knowledge property ensures that nothing is revealed about the secret key, and the argument of knowledge property forces the issuer to know the associated secret key. This way, the adversary must know the secret key associated to any authority it creates maliciously, since it has to provide an argument of knowledge. Note that in our ABE constructions we use a ZK-AoK for a very simple language that admits an efficient sigma protocol, that can be made non-interactive with the Fiat-Shamir heuristic. Consequently, strengthening the security comes at a modest efficiency cost. In the rest of this paper, we focus on the weaker security definition, which is easier to prove. Definitions regarding Zero-Knowledge are given in the Appendix B.

5 Modular Constructions of Multi-Authority ABE

We present several modular constructions of multi-authority ABE for access structures represented by monotone span programs, based on identity-based inner-product FE. This approach yields several schemes with various efficiency, security trade-offs.

The first construction relies on the multi-instance simulation security of the underlying FE, and assumes no a-priori bound on the number of attributes per authority used in the ciphertext or user secret keys. We present an FE that fulfills the security requirements, whose proof is given in the GGM. Note that randomness re-use yields an efficient two-step decryption which reduces the number of pairing operations required.

Then, we present a similar modular construction that only requires the simulation security of the underlying FE to hold in the single ciphertext setting. The advantage is we can build such FE from standard assumptions (in fact we sketch an impossibility result showing that the number of ciphertexts must be bounded in the simulation setting in the standard model), in the selective setting. The inconvenient is that the number of attributes in the access structure owned by a given authority must be a priori bounded, and the size of the user secret key grows (linearly) with that bound.

5.1 MA-ABE with an Unbounded Number of Attributes per Authority

We show that our modular construction from Fig. 3 yields an adaptively secure MA-ABE, against generic adversaries, when it is instantiated with the ID-based inner-product FE from Fig. 4, if the hash function is modeled as a random oracle.

Correctness. Let $[\![z_{\mathsf{gid}}]\!]_2 \coloneqq H(\mathsf{gid})$ and observe that, by the correctness of Γ:

$$\sum_{j=1}^{\ell} \omega_j \Gamma.\mathsf{Dec}(\mathsf{pk}_{\theta(j)}, \mathsf{ct}_j, \mathsf{sk}_{\rho(j),\mathsf{gid}}, [\![1, z_{\mathsf{gid}}]\!]_2) = \sum_{j=1}^{\ell} [\![\omega_j(s_j + u_j z_{\mathsf{gid}})]\!]_{\mathsf{t}} = [\![s + 0 z_{\mathsf{gid}}]\!]_{\mathsf{t}} = \kappa \ .$$

Optimized decryption. We show that the above computation for correctness, presented generically for any ID-IPFE Γ, can be performed significantly more efficiently by leveraging the structure of the ID-IPFE from Fig. 4 with randomness re-use. It favors exponentiations in source groups rather than pairing operations or exponentiations in the target group, which are more costly. Namely, for all $j \in [\ell]$, we have:

$$\mathsf{ct}_j = \left([\![r_{\xi(j)}]\!]_1 , \left[\!\!\left[(u_0^{\theta(j)} + \rho(j) u_1^{\theta(j)}) r_{\xi(j)} \right]\!\!\right]_1 , \left[\!\!\left[\boldsymbol{v}^{\theta(j)} r_{\xi(j)} + (s_j, u_j) \right]\!\!\right]_1 \right).$$

$\underline{\mathsf{GlobalSetup}(1^\lambda):}$

Run $\Gamma.\mathsf{gp} \leftarrow \Gamma.\mathsf{GlobalSetup}(1^\lambda)$, $\Gamma.\mathsf{gp}$ defines a pairing group $(p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathbb{G}_\mathsf{t}, e)$. Set a hash function $H : \{0,1\}^* \to \mathbb{G}_2$ and return $\mathsf{gp} \coloneqq (\Gamma.\mathsf{gp}, H)$.

$\underline{\mathsf{AuthSetup}(\mathsf{gp}):}$

Return $(\mathsf{pk}, \mathsf{sk}) \leftarrow \Gamma.\mathsf{Setup}(\Gamma.\mathsf{gp})$.

$\underline{\mathsf{Enc}\big((\boldsymbol{M} \in \mathbb{Z}_p^{n\times\ell}, \rho : [\ell] \to \{0,1\}^*), \{\mathsf{pk}_i\}_{i\in[\nu]}\big):}$

Sample $s \leftarrow_R \mathbb{Z}_p$, and $\{s_j\}_{j\in[\ell]} \leftarrow \mathsf{Share}(\boldsymbol{M}, s)$, $\{u_j\}_{j\in[\ell]} \leftarrow \mathsf{Share}(\boldsymbol{M}, 0)$.
Sample $r_1, \ldots, r_t \leftarrow_R \{0,1\}^*$ where $t = \max_{j\in[\ell]}\{\xi(j)\}$.
For all $j \in [\ell]$, $\mathsf{ct}_j \coloneqq \Gamma.\mathsf{Enc}(\mathsf{pk}_{\theta(j)}, (s_j, u_j), \rho(j); r_{\xi(j)})$.
Return $\big(\{\mathsf{ct}_j\}_{j\in[\ell]}, \kappa \coloneqq [\![s]\!]_\mathsf{t}\big)$.

$\underline{\mathsf{KeyGen}(\mathsf{sk}, \mathsf{gid}, \mathsf{a}):}$

Write $H(\mathsf{gid}) = [\![z_\mathsf{gid}]\!]_2$. Return $\mathsf{sk}_{\mathsf{a},\mathsf{gid}} \leftarrow \Gamma.\mathsf{KeyGen}(\mathsf{sk}, [\![1, z_\mathsf{gid}]\!]_2, \mathsf{id} = \mathsf{a})$.

$\underline{\mathsf{Dec}\left(\mathsf{ct}_{M,\rho} \coloneqq \{\mathsf{ct}_j\}_{j\in[\ell]}, \{\mathsf{sk}_\mathsf{a}\}_{\mathsf{a}\in\mathcal{S}}\right):}$

Compute $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_p$ such that $\sum_{j\in[\ell]} \omega_j \boldsymbol{M}_j = \mathbf{1} \wedge \omega_j = 0$ if $\rho(j) \notin \mathcal{S}$
Return $\sum_{j=1}^{\ell} \omega_j \Gamma.\mathsf{Dec}(\mathsf{pk}_{\theta(j)}, \mathsf{ct}_j, \mathsf{sk}_{\rho(j)}, [\![1, z_\mathsf{gid}]\!]_2)$

Fig. 3. Multi-Authority ABE from IB-IPFE. Here, Γ is an identity-based inner product functional encryption scheme (for 2-dimensional vectors). Recall that θ maps a row $j \in [\ell]$ to the authority that owns the attribute associated to that row, and ξ is just used to number the attributes owned by a given authority.

For all gid and all $j \in [\ell]$, we have:

$$\mathsf{sk}_{\rho(j),\mathsf{gid}} = \left([\![s_{\rho(j),\mathsf{gid}}]\!]_2 , \left[\!\!\left[\boldsymbol{v}^{\theta(j)\top}(1, z_\mathsf{gid}) + (u_0^{\theta(j)} + \rho(j) u_1^{\theta(j)}) s_{\rho(j),\mathsf{gid}} \right]\!\!\right]_2 \right).$$

Optimized decryption computes the following.

- First step: $[\![\boldsymbol{c}]\!]_1 = \sum_{j=1}^{\ell} \omega_j [\![\boldsymbol{v}^{\theta(j)} r_{\xi(j)} + (s_j, u_j)]\!]_1$. For all $m \in [t]$,

$$[\![k_m]\!]_2 = \sum_{j:\xi(j)=l} \omega_j \left[\!\!\left[\boldsymbol{v}^{\theta(j)\top}(1, z_\mathsf{gid}) + (u_0^{\theta(j)} + \rho(j) u_1^{\theta(j)}) s_{\rho(j),\mathsf{gid}} \right]\!\!\right]_2 .$$

- Second step: $[\![d_0]\!]_\mathsf{t} = e([\![\boldsymbol{c}]\!]_1^\top, [\![1, z_\mathsf{gid}]\!]_2^\top)$. For all $m \in [t]$, $[\![d_{1,m}]\!]_\mathsf{t} = e([\![r_m]\!]_1, [\![k_m]\!]_2)$. For all $j \in [\ell]$, $[\![d_{2,j}]\!]_\mathsf{t} = e(\left[\!\!\left[(u_0^{\theta(j)} + \rho(j) u_1^{\theta(j)}) r_{\xi(j)} \right]\!\!\right]_1, [\![s_{\rho(j),\mathsf{gid}}]\!]_2)^{\omega_j}$. It returns $[\![d_0]\!]_\mathsf{t} - \sum_{m\in[t]} [\![d_{1,m}]\!]_\mathsf{t} + \sum_{j\in[\ell]} [\![d_{2,j}]\!]_\mathsf{t}$.

The total cost of the optimized decryption are given in Table 1.

Theorem 2. *The scheme from Fig. 3, instantiated with the ID-based inner-product FE from Fig. 4, is an adaptively secure multi-authority CP-ABE in the generic group, random oracle model.*

Proof. Consider the adaptive security game for multi-authority ABE, defined in Section 4. We focus on its symbolic security, which implies its security in the generic group model (some background on the symbolic and generic group model is given in the Appendix A). Note that the global setup consists of sampling a pairing group $\Gamma.\mathsf{gp} \coloneqq (p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathbb{G}_\mathsf{t}, e)$ for the given security parameter $\lambda \in \mathbb{N}$. In the symbolic security experiment, group elements will be represented by polynomials stored in three lists, one corresponding to each of the groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_\mathsf{t}$.

$\underline{\mathsf{GlobalSetup}(1^\lambda)}$:

Given $\lambda \in \mathbb{N}$, sample $\mathcal{PG} := (p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathbb{G}_\mathsf{t}, e) \leftarrow \mathsf{GGen}(1^\lambda)$ and output $\mathsf{gp} = \mathcal{PG}$.

$\underline{\mathsf{Setup}(\mathsf{gp})}$:

Given gp, sample $u_0, u_1 \leftarrow_R \mathbb{Z}_p$, $\boldsymbol{v} \leftarrow_R \mathbb{Z}_p^d$.
Set $\mathsf{msk} = (\boldsymbol{v}, u_0, u_1)$, $\mathsf{pk} = ([\![\boldsymbol{v}]\!]_1, [\![u_0]\!]_1, [\![u_1]\!]_1)$ and output $(\mathsf{msk}, \mathsf{pk})$.

$\underline{\mathsf{Enc}(\mathsf{pk}, \boldsymbol{x}, \mathsf{id};\ r)}$:

Given pk, $\boldsymbol{x} \in \mathbb{Z}_p^d$, $\mathsf{id} \in \mathbb{Z}_p$, and random coins $r \in \mathbb{Z}_p$.
Return $\mathsf{ct} = ([\![r]\!]_1, [\![(u_0 + \mathsf{id}u_1)r]\!]_1, [\![\boldsymbol{v}r + \boldsymbol{x}]\!]_1) \in \mathbb{G}_1^{2+d}$.

$\underline{\mathsf{KeyGen}(\mathsf{msk}, [\![\boldsymbol{y}]\!]_2 \in \mathbb{G}_2^d, \mathsf{id} \in \mathbb{Z}_p)}$:

Given msk, sample $s \leftarrow_R \mathbb{Z}_p$.
Return $\mathsf{sk} = ([\![s]\!]_2, [\![\boldsymbol{v}^\top \boldsymbol{y} + (u_0 + \mathsf{id}u_1)s]\!]_2, [\![\boldsymbol{y}]\!]_2) \in \mathbb{G}_2^{2+d}$.

$\underline{\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}, [\![\boldsymbol{y}]\!]_2)}$:

Parse ct as $([\![c_1]\!]_1, [\![c_2]\!]_1, [\![\boldsymbol{c}_3]\!]_1 \in \mathbb{G}_1^d)$ and sk as $([\![k_1]\!]_2, [\![k_2]\!]_2)$.
Return $e([\![\boldsymbol{c}_3^\top]\!]_1, [\![\boldsymbol{y}]\!]_2) \cdot e([\![c_2]\!]_1, [\![k_1]\!]_2) \,/\, e([\![c_1]\!]_1, [\![k_2]\!]_2)$.

Fig. 4. ID-based inner-product FE for d-dimensional vectors and ID space $= \mathbb{Z}_p$. We directly prove its security in combination with our modular construction of MA-ABE from Fig. 3, in the GGM.

On the i-th authority creation query, a pair of keys will be created following $\Gamma.\mathsf{Setup}(\Gamma.\mathsf{gp})$. In particular, consider new formal variables $V_0^{(i)}, V_1^{(i)}, U_0^{(i)}, U_1^{(i)}$ representing the random values during the ID-IPFE key generation. The adversary gets handles to each of these variables as polynomials in $\mathbb{G}_1$.

On the i-th key generation query, say with respect to authority A_i and on input $(\mathsf{gid}_i, \mathsf{a}_i)$ such that $\mathsf{a}_i \in \mathcal{U}_{A_i} \subset \mathbb{Z}_p$, let Z_{gid_i} be the formal variable associated to value $H(\mathsf{gid}_i)$. Let S_i be a new formal variable corresponding to the sampled randomness during $\Gamma.\mathsf{KeyGen}$. The adversary will be given handles to polynomials S_i as well as sk_i (all in $\mathbb{G}_2$), defined as:

$$\mathsf{sk}_i := V_0^{(A_i)} + V_1^{(A_i)} Z_{\mathsf{gid}_i} + (U_0^{(A_i)} + \mathsf{a}_i U_1^{(A_i)}) S_i \ .$$

Eventually, the adversary will perform an encryption query, say on policy $\boldsymbol{M} \in \mathbb{Z}_p^{n \times \ell}, \rho : [\ell] \to \{0,1\}^*$, and for a chosen set of public keys Π (let $\Pi_{\mathsf{corr}} \subset \Pi$ be the set of corrupted authorities). Let S^* and $R_1, \ldots, R_t$, for $t := \max_{j \in [\ell]}\{\xi(j)\}$, and $T_2, \ldots, T_n$, $W_2, \ldots, W_n$ be new formal variables corresponding to the randomness generated during $\Gamma.\mathsf{Enc}$. The adversary will be given handles to polynomials $R_1, \ \ldots, \ R_t$ as well as the following (all in $\mathbb{G}_1$):

$$\begin{aligned}
\mathsf{ct}_j &:= (U_0^{(\theta(j))} + \rho(j) U_1^{(\theta(j))}) R_{\xi(j)} && \forall j \in [\ell] : \theta(j) \notin \Pi_{\mathsf{corr}} \\
\mathsf{ct}'_j &:= V_0^{(\theta(j))} R_{\xi(j)} + \textstyle\sum_{k=2}^n M_{k,j} T_k + M_{1,j} S^* && \forall j \in [\ell] : \theta(j) \notin \Pi_{\mathsf{corr}} \\
\mathsf{ct}''_j &:= V_1^{(\theta(j))} R_{\xi(j)} + \textstyle\sum_{k=2}^n M_{k,j} W_k && \forall j \in [\ell] : \theta(j) \notin \Pi_{\mathsf{corr}} \ ,
\end{aligned}$$

where $M_{k,j}$ represents the element of $\boldsymbol{M}$ in the k-th row and j-th column. Furthermore, for every $j \in [\ell]$ such that $\theta(j) \in \Pi_{\mathsf{corr}}$, the adversary will get handles to polynomials in the same form as above, with the exception that variables $U_0^{\theta(j)}$, $U_1^{\theta(j)}$, $V_0^{\theta(j)}$ and $V_1^{\theta(j)}$ will be replaced by values $\mathbb{Z}_p$ chosen by the adversary. However, note that the adversary has handles to all R_m, for $m \in [t]$, so polynomials ct_j corresponding to a corrupted j do not add symbolic power to the adversary. A similar reasoning allows us to conclude that we can focus on the case where the adversary gets handles to the following polynomials in the case of corrupted

authorities (we define $\mathsf{ct}_j \coloneqq 0$ if $\theta(j) \in \Pi_{\mathsf{corr}}$):

$$\begin{aligned} \mathsf{ct}'_j &\coloneqq \textstyle\sum_{k=2}^{n} M_{k,j} T_k + M_{1,j} S^* && \forall j \in [\ell] : \theta(j) \in \Pi_{\mathsf{corr}} \\ \mathsf{ct}''_j &\coloneqq \textstyle\sum_{k=2}^{n} M_{k,j} W_k && \forall j \in [\ell] : \theta(j) \in \Pi_{\mathsf{corr}} \ . \end{aligned}$$

The adversary will win the symbolic experiment if it can produce a handle pointing to polynomial S^* in $\mathbb{G}_{\mathsf{t}}$, which corresponds to the encapsulation key. In particular, let L_1 and L_2 be lists (or vectors) containing the polynomials that the adversary is given access to, in $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, at the end of its interaction with the oracles (note that the adversary is not directly given any polynomial in $\mathbb{G}_{\mathsf{t}}$). We have:

$$\begin{aligned} \mathsf{L}_1 &= \big(1,\ \{V_b^{(i)},\ U_b^{(i)}\}_{i\in[q_A],b\in\{0,1\}},\ \{R_m\}_{m\in[t]}, \{\mathsf{ct}_j,\ \mathsf{ct}'_j,\ \mathsf{ct}''_j\}_{j\in[\ell]}\big) \\ \mathsf{L}_2 &= \big(1,\ \{Z_{\mathsf{gid}_i},\ S_i,\ \mathsf{sk}_i\}_{i\in[q_{\mathsf{sk}}]}\big) \ , \end{aligned}$$

where q_A denotes the total number of authorities that were created and q_{sk} denotes the total number of key generation queries. Also, note that although the random oracle gives the adversary access to other monomials of the form Z_x for an arbitrary $x \in \{0,1\}^*$, only those corresponding to a requested gid are relevant for the experiment. We will show that no linear combination of the polynomials in $\mathsf{L}_{\mathsf{t}} \coloneqq \mathsf{L}_1 \otimes \mathsf{L}_2$ (representing the polynomials in $\mathbb{G}_{\mathsf{t}}$ that the adversary can get access to) can be equal to polynomial S^*, for any value of $(\mathsf{gid}_i, \mathsf{a}_i)$ for all $i \in [q_{\mathsf{sk}}]$ and policy $(\boldsymbol{M}, \rho)$ that make the adversary admissible.

Notice that the only polynomials which contain monomial S^* are in the form of $\mathsf{ct}'_j \cdot 1$ for $j \in [\ell]$. This means that the linear combination producing S^* must contain a summand of the following form, for certain coefficients $\omega_j \in \mathbb{Z}_p$:

$$\begin{aligned} \textstyle\sum_{j\in[\ell]} \omega_j \, \mathsf{ct}'_j \ &= \ \textstyle\sum_{j\in[\ell]:\theta(j)\notin\Pi_{\mathsf{corr}}} \omega_j \big(V_0^{(\theta(j))} R_{\xi(j)} + \sum_{k=2}^{n} M_{k,j} T_k + M_{1,j} S^*\big) \\ &\quad + \textstyle\sum_{j\in[\ell]:\theta(j)\in\Pi_{\mathsf{corr}}} \omega_j \big(\sum_{k=2}^{n} M_{k,j} T_k + M_{1,j} S^*\big) \ . \end{aligned} \tag{2}$$

Furthermore, observe that monomials T_k, for $k = 2, \ldots, n$ do not appear in any other polynomial. Consequently, for all $k \in [2, n]$, we must have:

$$\textstyle\sum_{j\in[\ell]} \omega_j M_{1,j} = 1 \quad \text{and} \quad \forall k \in [2,n],\ \sum_{j\in[\ell]} \omega_j M_{k,j} = 0 \ ,$$

or, more compactly, $\boldsymbol{M}\boldsymbol{\omega} = \mathbf{1}$. Also, notice that the linear combination from (2) leaves the following residue:

$$\textstyle\sum_{j\in[\ell]:\theta(j)\notin\Pi_{\mathsf{corr}}} \omega_j V_0^{(\theta(j))} R_{\xi(j)} \ , \tag{3}$$

which needs to be cancelled out with other available polynomials. However, monomials of the form $V_0^{(A)} R_m$ for some non-corrupted authority A and some $m \in [t]$, only appear in polynomials (from L_{t}) in the form of $R_m \cdot \mathsf{sk}_i$ for $i \in [q_{\mathsf{sk}}]$ such that $A_i = A$. The only way the above residue can be cancelled is by adding the following term to it, for some coefficients $\delta_{m,i} \in \mathbb{Z}_p$:

$$\textstyle\sum_{m\in[t],i\in[q_{\mathsf{sk}}]:A_i\notin\Pi_{\mathsf{corr}}} \delta_{m,i} \, R_m \cdot \big(V_0^{(A_i)} + V_1^{(A_i)} Z_{\mathsf{gid}_i} + (U_0^{(A_i)} + \mathsf{a}_i U_1^{(A_i)}) S_i\big) \ , \tag{4}$$

Furthermore, our randomness re-use policy (no re-use across the same authority) implies that all monomials in residue (3) are different, for all j. For a fixed $j \in [\ell] : \theta(j) \notin \Pi_{\mathsf{corr}}$, by focusing on the coefficients of monomial $V_0^{(\theta(j))} R_{\xi(j)}$, we conclude that $w_j + \sum_{i:A_i=\theta(j)} \delta_{\xi(j),i} = 0$ $(\star)$. Also, note that term (4) has introduced the following new residues:

$$\textstyle\sum_{m\in[t],i\in[q_{\mathsf{sk}}]:A_i\notin\Pi_{\mathsf{corr}}} \big(\, \delta_{m,i} \, R_m V_1^{(A_i)} Z_{\mathsf{gid}_i} \ + \ \delta_{m,i} \, (U_0^{(A_i)} + \mathsf{a}_i U_1^{(A_i)}) R_m S_i \,\big) \ .$$

These residues can (only) be cancelled by adding the following terms, for some coefficients $\alpha_{j,i}, \beta_{j,i} \in \mathbb{Z}_p$:

$$\begin{aligned} \textstyle\sum_{j\in[\ell]:\theta(j)\notin\Pi_{\mathsf{corr}},i\in[q_{\mathsf{sk}}]:A_i\notin\Pi_{\mathsf{corr}}} \alpha_{j,i} \big(V_1^{(\theta(j))} R_{\xi(j)} + \textstyle\sum_{k=2}^{n} M_{k,j} W_k\big) Z_{\mathsf{gid}_i} && [\mathsf{ct}''_j \cdot Z_{\mathsf{gid}_i}] \\ \textstyle\sum_{j\in[\ell]:\theta(j)\in\Pi_{\mathsf{corr}},i\in[q_{\mathsf{sk}}]:A_i\notin\Pi_{\mathsf{corr}}} \alpha_{j,i} \big(\textstyle\sum_{k=2}^{n} M_{k,j} W_k\big) Z_{\mathsf{gid}_i} && [\mathsf{ct}''_j \cdot Z_{\mathsf{gid}_i}] \\ \textstyle\sum_{j\in[\ell]:\theta(j)\notin\Pi_{\mathsf{corr}},i\in[q_{\mathsf{sk}}]:A_i\notin\Pi_{\mathsf{corr}}} \beta_{j,i} \, (U_0^{(\theta(j))} + \rho(j) U_1^{(\theta(j))}) R_{\xi(j)} S_i && [\mathsf{ct}_j \cdot S_i] \ , \end{aligned}$$

As before, for a fixed $j \in [\ell] : \theta(j) \notin \Pi_{\mathsf{corr}}$ and a fixed gid, by considering the coefficients of the following monomials we get:

$$V_1^{(\theta(j))} R_{\xi(j)} Z_{\mathsf{gid}} \quad \rightarrow \quad \sum_{i:\mathsf{gid}_i=\mathsf{gid}} \alpha_{j,i} + \sum_{i:A_i=\theta(j)\wedge \mathsf{gid}_i=\mathsf{gid}} \delta_{\xi(j),i} = 0 \tag{5}$$

$$U_0^{(\theta(j))} R_{\xi(j)} S_i \quad \rightarrow \quad \beta_{j,i} + \delta_{\xi(j),i} = 0 \qquad \forall i \in [q_{\mathsf{sk}}] : A_i = \theta(j) \tag{6}$$

$$U_1^{(\theta(j))} R_{\xi(j)} S_i \quad \rightarrow \quad \beta_{j,i}\, \rho(j) + \delta_{\xi(j),i}\, \mathsf{a}_i = 0 \qquad \forall i \in [q_{\mathsf{sk}}] : A_i = \theta(j) \ . \tag{7}$$

From equations (6) and (7) we can deduce that for every $j \in [\ell] : \theta(j) \notin \Pi_{\mathsf{corr}}$ and all $i \in [q_{\mathsf{sk}}]$ such that $A_i = \theta(j)$, we have $\delta_{\xi(j),i}\,(\rho(j) - \mathsf{a}_i) = 0$. Which means that $\delta_{\xi(j),i}$ must be zero unless $\rho(j) = \mathsf{a}_i$ in the case of non-corrupted authorities. Furthermore, from looking at the coefficient of $W_k Z_{\mathsf{gid}}$, for any $k \in [2, n]$, we get that $\sum_{j\in[\ell],i:\mathsf{gid}_i=\mathsf{gid}} \alpha_{j,i} M_{k,j} = 0$. Using these two facts over equation (5) and summing over all $j \in [\ell] : \theta(j) \notin \Pi_{\mathsf{corr}}$ (after multiplying it by $M_{k,j}$ for a fixed $k \in [2, n]$), we deduce that:

$$\begin{aligned} 0 \ = \ & \sum_{j\in[\ell]:\theta(j)\notin\Pi_{\mathsf{corr}},\, i:A_i=\theta(j)\,\wedge\,\rho(j)=a_i\,\wedge\,\mathsf{gid}_i=\mathsf{gid}} \delta_{\xi(j),i}\, M_{k,j} \\ & - \sum_{j\in[\ell]:\theta(j)\in\Pi_{\mathsf{corr}},\, i:\mathsf{gid}_i=\mathsf{gid}} \alpha_{j,i}\, M_{k,j} \qquad \forall k \in [2, n] \ . \end{aligned} \tag{8}$$

Note that, for any fixed gid, equation (8) represents a linear combination of columns of $\boldsymbol{M}$ that adds up to zero in all rows $k \in [2, n]$. Furthermore, it only involves columns corresponding to corrupted authorities or authorities for which the corresponding attribute has been requested (for the specified gid). We will now deduce that if such a combination exists, the adversary must be non-admissible. In particular, we will show that there must be at least one gid for which the above linear combination of columns is non zero on the first row (corresponding to $k = 1$), concluding the proof.

Consider the following equation, relative to the right-hand side of equation (8), with 1 instead of k and having added up over all gid:

$$\sum_{j\in[\ell]:\theta(j)\notin\Pi_{\mathsf{corr}},\, i:A_i=\theta(j)\,\wedge\,\rho(j)=a_i} \delta_{\xi(j),i}\, M_{1,j} - \sum_{j\in[\ell]:\theta(j)\in\Pi_{\mathsf{corr}},\, i\in[q_{\mathsf{sk}}]} \alpha_{j,i}\, M_{1,j} \ . \tag{9}$$

We conclude the proof by arguing that the value of the expression in equation (9) must be non-zero. This implies that it must also be non-zero when projected to at least one of the gid's, as desired. First recall that our equation ($\star$), updated with our new information about $\delta_{\xi(j),i}$ (when $\rho(j) \neq \mathsf{a}_i$) tells us that for every $j \in [\ell] : \theta(j) \notin \Pi_{\mathsf{corr}}$,

$$\sum_{i:A_i=\theta(j)\wedge\rho(j)=a_i} \delta_{\xi(j),i} = -\omega_j \ ,$$

which implies that, for every $k \in [n]$ (multiply by $M_{k,j}$ and sum over j):

$$\sum_{j\in[\ell]:\theta(j)\notin\Pi_{\mathsf{corr}},i:A_i=\theta(j)\wedge\rho(j)=a_i} \delta_{\xi(j),i} M_{k,j} = -\sum_{j:\theta(j)\notin\Pi_{\mathsf{corr}}} \omega_j M_{k,j} \ , \tag{10}$$

Finally, applying equation (10) on expression (9); using equation (8) for all $k \in [2, n]$, and leveraging the fact that $\boldsymbol{M}\boldsymbol{\omega} = \mathbf{1}$, we can express (9) as -1 plus a linear combination of only corrupted columns that has all components $k = 1, \ldots, n$ equal to zero, and thus, vanishes. This implies that expression (9) is non-zero, as desired. □

5.2 MA-ABE with a Bounded Number of Attributes per Authority in the Ciphertexts

We present a modular construction in Fig. 5 that is similar to the one in Fig. 3 except this time, we assume the encryptor only uses access structures where the number of attributes owned by a given authority is a-priori bounded. It uses as underlying building blocks an ID-based inner-product FE. When instantiated with the ID-based inner-product FE from Fig. 2 with vectors of dimension $d = 2$, it yields a selectively secure MA-ABE from standard assumptions in the random oracle model.

$\underline{\mathsf{GlobalSetup}(1^\lambda):}$
Run $\Gamma.\mathsf{gp} \leftarrow \Gamma.\mathsf{GlobalSetup}(1^\lambda)$, $\Gamma.\mathsf{gp}$ defines a pairing group $(p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathbb{G}_\mathsf{t}, e)$.
Set a hash function $H : \{0,1\}^* \to \mathbb{G}_2$ and return $\mathsf{gp} \coloneqq (\Gamma.\mathsf{gp}, H)$.

$\underline{\mathsf{AuthSetup}(\mathsf{gp}):}$
For all $i \in [B]$, compute $(\Gamma.\mathsf{pk}^i, \Gamma.\mathsf{msk}^i) \leftarrow \Gamma.\mathsf{Setup}(\Gamma.\mathsf{gp})$.
Set $\mathsf{pk} = (\Gamma.\mathsf{pk}^1, \ldots, \Gamma.\mathsf{pk}^B)$ and $\mathsf{sk} = (\Gamma.\mathsf{msk}^1, \ldots, \Gamma.\mathsf{msk}^B)$.
Return $(\mathsf{pk}, \mathsf{sk})$.

$\underline{\mathsf{Enc}(\boldsymbol{M} \in \mathbb{Z}_p^{n \times \ell}, \rho : [\ell] \to \mathbb{Z}_p, \{\mathsf{pk}_1, \ldots, \mathsf{pk}_\nu\}):}$
For all $t \in [\nu]$, parse $\mathsf{pk}_t = (\Gamma.\mathsf{pk}_t^1, \ldots, \Gamma.\mathsf{pk}_t^B)$.
Sample $s \leftarrow_R \mathbb{Z}_p$, $\{s_j\}_{j\in[\ell]} \leftarrow \mathsf{Share}(\boldsymbol{M}, s)$, $\{u_j\}_{j\in[\ell]} \leftarrow \mathsf{Share}(\boldsymbol{M}, 0)$, $r \leftarrow_R \mathcal{R}$.
Set $\Gamma.\mathsf{ct}_j \coloneqq \Gamma.\mathsf{Enc}\big(\Gamma.\mathsf{pk}_{\theta(j)}^{\xi(j)}, (s_j, u_j), \rho(j);\, r\big)$, for all $j \in [\ell]$, $\theta(j) \in [\nu]$.
Return $\big(\{\Gamma.\mathsf{ct}_j\}_{j\in[\ell]}, \kappa \coloneqq [\![s]\!]_\mathsf{t}\big)$.

$\underline{\mathsf{KeyGen}\big(\mathsf{sk}, \mathsf{gid}, \mathsf{a}\big):}$
Parse $H(\mathsf{gid}) = [\![z_\mathsf{gid}]\!]_2$ and $\mathsf{sk} = (\Gamma.\mathsf{msk}^1, \ldots, \Gamma.\mathsf{msk}^B)$.
For all $i \in [B]$, set $\Gamma.\mathsf{sk}^i \leftarrow \Gamma.\mathsf{KeyGen}(\Gamma.\mathsf{msk}^i, [\![1, z_\mathsf{gid}]\!]_2, \mathsf{a})$.
Return $\mathsf{sk}_{\mathsf{gid},\mathsf{a}} = (\Gamma.\mathsf{sk}^1, \ldots, \Gamma.\mathsf{sk}^B)$.

$\underline{\mathsf{Dec}\left(\mathsf{ct}_{(\boldsymbol{M},\rho)}, \{\mathsf{sk}_{\mathsf{gid},\mathsf{a}}\}_{\mathsf{a}\in\mathcal{S}}\right):}$
Compute $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_p$ such that $\sum_{j\in[\ell]} \omega_j \boldsymbol{M}_j = \mathbf{1} \wedge \omega_j = 0$ if $\rho(j) \notin \mathcal{S}$.
Parse $\mathsf{ct} = \{\Gamma.\mathsf{ct}_j\}_{j\in[\ell]}$ and for all $\mathsf{a} \in \mathcal{S}$, parse $\mathsf{sk}_{\mathsf{gid},\mathsf{a}} = (\Gamma.\mathsf{sk}_\mathsf{a}^1, \ldots, \Gamma.\mathsf{sk}_\mathsf{a}^B)$.
For all $j \in [\ell]$ s.t. $\rho(j) \in \mathcal{S}$, compute $[\![\gamma_j]\!]_\mathsf{t} = \Gamma.\mathsf{Dec}\big(\Gamma.\mathsf{ct}_j, \Gamma.\mathsf{sk}_{\rho(j)}^{\xi(j)}, [\![1, z_\mathsf{gid}]\!]_2\big)$.
Return $\sum_{j=1}^{\ell} \omega_j [\![\gamma_j]\!]_\mathsf{t}$.

Fig. 5. MA-ABE from ID-IPFE for a bounded number of attributes per authority in the ciphertexts. Here, Γ is the ID-based inner-product FE from Fig. 2, for 2-dimensional vectors. Recall that θ maps a row $j \in [\ell]$ to the authority that owns the attribute $\rho(j)$ associated to that row, and ξ is just used to number the attributes owned by a given authority. Finally, B upper bounds the number of attributes per authority. That is, for all $j \in [\ell]$, $\xi(j) \leq B$.

Correctness. By correctness of the inner-product FE Γ, for all $j \in [\ell]$ such that $\rho(j) \in \mathcal{S}$, we have $\Gamma.\mathsf{Dec}\big(\Gamma.\mathsf{ct}_j, \Gamma.\mathsf{sk}_{\rho(j)}^{\xi(j)}\big) = [\![s_j + u_j z_\mathsf{gid}]\!]_\mathsf{t}$. By correctness of the access structure, $\sum_{j=1}^{\ell} \omega_j [\![s_j + u_j z_\mathsf{gid}]\!]_\mathsf{t} = [\![s + 0 \cdot z_\mathsf{gid}]\!]_\mathsf{t} = [\![s]\!]_\mathsf{t}$.

Optimized decryption. When the underlying Γ is the ID-IPFE from Fig. 2, we can use an optimized, two-step decryption to minimize the number of costly pairing operations and exponentiations in the target group, by doing some pre-computations that involve less costly exponentiations in source groups. This relies on the structure of the ID-IPFE from Fig. 2 with randomness re-use. Namely, for all $j \in [\ell]$, we have:

$$\mathsf{ct}_j = \left([\![\boldsymbol{a}r]\!]_1, \left[\!\!\left[(\boldsymbol{U}_{\theta(j),0}^{\xi(j)} + \rho(j)\boldsymbol{U}_{\theta(j),1}^{\xi(j)})\boldsymbol{a}r \right]\!\!\right]_1, \left[\!\!\left[\boldsymbol{V}_{\theta(j)}^{\xi(j)}\boldsymbol{a}r + (s_j, u_j)^\top \right]\!\!\right]_1 \right).$$

For all gid and all $j \in [\ell]$, we have:

$$\mathsf{sk}_{\rho(j),\mathsf{gid}} = \left(\left[\!\!\left[\boldsymbol{b}s_{\rho(j),\mathsf{gid}}^m \right]\!\!\right]_2, \left[\!\!\left[\boldsymbol{V}_{\theta(j)}^{m\top}(1, z_\mathsf{gid}) + (\boldsymbol{U}_{\theta(j),0}^{\xi(j)} + \rho(j)\boldsymbol{U}_{\theta(j),1}^{\xi(j)})^\top \boldsymbol{b}s_{\rho(j),\mathsf{gid}}^m \right]\!\!\right]_2 \right)_{m\in[B]}.$$

Optimized decryption computes the following.

- First step: $[\![\boldsymbol{c}]\!]_1 = \sum_{j=1}^{\ell} \omega_j \left[\!\!\left[\boldsymbol{V}_{\theta(j)}^{\xi(j)} \boldsymbol{a}r + (s_j, u_j)^\top \right]\!\!\right]_1$. For all $m \in [B]$,

$$[\![\boldsymbol{k}_m]\!]_2 = \sum_{j=1}^{\ell} \omega_j \left[\!\!\left[\boldsymbol{V}_{\theta(j)}^{m\top}(1, z_{\mathsf{gid}}) + (\boldsymbol{U}_{\theta(j),0}^{\xi(j)} + \rho(j)\boldsymbol{U}_{\theta(j),1}^{\xi(j)})^\top \boldsymbol{b}s_{\rho(j),\mathsf{gid}}^{m} \right]\!\!\right]_2 .$$

- Second step: $[\![d_0]\!]_{\mathsf{t}} = e([\![\boldsymbol{c}]\!]_1^\top, [\![1, z_{\mathsf{gid}}]\!]_2^\top)$. For all $m \in [t]$, $[\![d_{1,m}]\!]_{\mathsf{t}} = e([\![\boldsymbol{a}r]\!]_1^\top, [\![\boldsymbol{k}_m]\!]_2)$. For all $j \in [\ell]$, $[\![d_{2,j}]\!]_{\mathsf{t}} = e(\left[\!\!\left[(\boldsymbol{U}_{\theta(j),0}^{\xi(j)} + \rho(j)\boldsymbol{U}_{\theta(j),1}^{\xi(j)})\boldsymbol{a}r \right]\!\!\right]_1^\top, \left[\!\!\left[\boldsymbol{b}s_{\rho(j),\mathsf{gid}}^{\xi(j)} \right]\!\!\right]_2)^{\omega_j}$. It returns $[\![d_0]\!]_{\mathsf{t}} - \sum_{j\in[\ell]} [\![d_{1,\xi(j)}]\!]_{\mathsf{t}} + \sum_{j\in[\ell]} [\![d_{2,j}]\!]_{\mathsf{t}}$.

The total cost of the optimized decryption are give in Table 1.

Theorem 3 (Security). *For any ID-based inner-product FE scheme Γ that is multi-instance simulation selectively secure, the scheme from Figure 5 that builds upon Γ is selectively secure in the random oracle model.*

Proof (Security). We prove security via a sequence of hybrid games.

$\underline{\mathsf{Game}_0}$: The first game corresponds to the selective security game for MA-ABE, with static corruptions, defined in Section 4. We recall it here for completeness. We call $\mathcal{A}$ the admissible adversary for the selective security of the MA-ABE scheme. First, $\mathcal{A}$ receives the global parameters $\mathsf{gp} = (\Gamma.\mathsf{gp}, H)$. Then, it can query its oracle $\mathcal{O}_{\mathsf{create}}$ that creates a new (honest) authority with an associated $(\mathsf{pk}, \mathsf{sk})$ pair when invoked, and returns pk to $\mathcal{A}$. We have $\mathsf{sk} = (\Gamma.\mathsf{msk}^1, \ldots, \Gamma.\mathsf{msk}^B)$ and $\mathsf{pk} = (\Gamma.\mathsf{pk}^1, \ldots, \Gamma.\mathsf{pk}^B)$. Then, $\mathcal{A}$ sends $(\boldsymbol{M}, \rho, \Pi_{\mathsf{hon}}, \Pi_{\mathsf{corr}})$ to its challenger, where $\boldsymbol{M} \in \mathbb{Z}_p^{n\times\ell}$, $\rho : [\ell] \to \mathbb{Z}_p$ is an access structure with attributes owned by the authorities in the set Π. We write $\Pi = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_\nu\}$, and we define $\theta : [\ell] \to [\nu]$, which maps each column $j \in [\ell]$ to the authority that owns the attribute associated with that column. Because several attributes used in $\boldsymbol{M}$ can be owned by the same authority, recall that we define a map $\xi : [\ell] \to [B]$ that maps each column $j \in [\ell]$ to an ordinal number such that $\rho(j)$ is the $\xi(j)$'th attribute used in the access structure owned by authority $\theta(j)$, sorted according to some linear ordering on the columns.

Upon receiving $(\boldsymbol{M}, \rho, \Pi)$, the challenger samples $s \leftarrow_R \mathbb{Z}_p$, and computes $(s_1, \ldots, s_\ell) \leftarrow \mathsf{Share}(\boldsymbol{M}, s)$, $(u_1, \ldots, u_\ell) \leftarrow \mathsf{Share}(\boldsymbol{M}, 0)$[4], $\kappa_0 = [\![s]\!]_{\mathsf{t}}$, $\kappa_1 \leftarrow_R \mathbb{G}_{\mathsf{t}}$, $b \leftarrow_R \{0,1\}$, $\kappa = \kappa_b$, $r \leftarrow_R \mathbb{Z}_p$, for all $j \in [\ell]$,

$$\Gamma.\mathsf{ct}_j = \Gamma.\mathsf{Enc}\left(\Gamma.\mathsf{pk}_{\theta(j)}^{\xi(j)}, (s_j, u_j), \rho(j); r\right), \quad \mathsf{ct}^\star = \{\Gamma.\mathsf{ct}_j\}_{j\in[\ell]} ,$$

and returns $(\mathsf{ct}^\star, \kappa)$ to $\mathcal{A}$. Note that the same randomness $r \leftarrow_R \mathbb{Z}_p$ is used to compute every $\Gamma.\mathsf{ct}_j$. By the decomposablility property of the FE scheme Γ, we have $\Gamma.\mathsf{ct}_j = (\mathsf{hd}, \mathsf{pl})$, where the header hd does not depend on (s_j, u_j) or $\rho(j)$, and a payload pl that can be computed deterministically from hd and the secret keys of the authority $\theta(j)$. Note that some of the authorities in the set Π may be created by $\mathcal{A}$ itself (and not via $\mathcal{O}_{\mathsf{create}}$); these are referred to as corrupted authorities, whereas the authorities created via $\mathcal{O}_{\mathsf{create}}$ are called honest. Also note that $\mathcal{A}$ cannot query its oracle $\mathcal{O}_{\mathsf{KeyGen}}$ before receiving the challenge ciphertext $(\mathsf{ct}^\star, \kappa)$ since we are in the selective setting.

The adversary $\mathcal{A}$ can then query its oracle $\mathcal{O}_{\mathsf{KeyGen}}$, which given as input $(\mathsf{pk}, \mathsf{a}, \mathsf{gid})$ where pk is an honest authority associated with secret key $\mathsf{sk} = (\Gamma.\mathsf{msk}^1, \ldots, \Gamma.\mathsf{msk}^B)$, returns $(\Gamma.\mathsf{sk}_{\mathsf{a}}^1, \ldots, \Gamma.\mathsf{sk}_{\mathsf{a}}^B) \leftarrow \mathsf{KeyGen}(\mathsf{sk}, \mathsf{a}, \mathsf{gid})$, where for all $i \in [B]$, $\Gamma.\mathsf{sk}_{\mathsf{a}}^i \leftarrow \Gamma.\mathsf{KeyGen}(\Gamma.\mathsf{msk}^i, [\![1, z_{\mathsf{gid}}]\!]_2, \mathsf{a})$,with $H(\mathsf{gid}) = [\![z_{\mathsf{gid}}]\!]_2$. It can also query its oracle $\mathcal{O}_{\mathsf{create}}$ again which has the same effect as before. Finally, $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. Recall that $\mathcal{A}$ is admissible which means it cannot compute κ_0 from $\mathsf{ct}^\star$ simply by correctness of the scheme with the user secret keys it queried and the secret key of the corrupted authorities (see Section 4 for more details). The experiment outputs 1 if $b = b'$, 0 otherwise.

$\underline{\mathsf{Game}_1}$: is the same as Game_0 except that the challenge ciphertext is computed as follows. Upon receiving $(\boldsymbol{M}, \rho, \Pi)$, the challenger computes $\mathsf{ct}^\star = \{\Gamma.\mathsf{ct}_j\}_{j\in[\ell]}$, where for all $j \in [\ell]$, if $\theta(j)$ is honest, then

$$\Gamma.\mathsf{ct}_j = \Gamma.\mathsf{Enc}\left(\Gamma.\mathsf{pk}_{\theta(j)}^{\xi(j)}, \rho(j), (s_j, u_j); r\right) = (\mathsf{hd}, \mathsf{pl}) ,$$

[4] See Fig. 1 for the definition of the algorithm Share.

as in Game_0. By the decomposablility property of Γ (see the definition in Section 3.1), we know that the header hd does not depend on the identity $\rho(j)$ or the vector (s_j, u_j), i.e. $\mathsf{hd} = \Gamma.\mathsf{Enc}\left(\Gamma.\mathsf{pk}_{\theta(j)}^{\xi(j)}; r\right)$. Since the same randomness r is used to compute all ciphertexts $\Gamma.\mathsf{ct}_j$ for all $j \in [\ell]$, this defines a unique header hd.

If $\theta(j)$ is corrupted instead of computing the ciphertext $\Gamma.\mathsf{ct}_j$ using the algorithm $\Gamma.\mathsf{Enc}$, it computes $\mathsf{pl} = \Gamma.\mathsf{Enc}_{\mathsf{pl}}(\Gamma.\mathsf{sk}_{\theta(j)}^{\xi(j)}, \rho(j), (s_j, u_j), \mathsf{hd})$, which can be computed since the adversary is restricted to provide the secret key associated to corrupted authorities. Finally it outputs $\mathsf{ct}_j = (\mathsf{hd}, \mathsf{pl})$. Consequently, given the decomposablility property of Γ, we have $\mathsf{Game}_0 = \mathsf{Game}_1$.

$\underline{\mathsf{Game}_2}$: is the same as Game_1 except that the keys and ciphertexts from the honest authorities are now computed using a simulator for Γ. Namely, the global parameters $\mathsf{gp} = (\Gamma.\overline{\mathsf{gp}}, H)$, where $(\Gamma.\overline{\mathsf{gp}}, \Gamma.\mathsf{td}) \leftarrow \Gamma.\overline{\mathsf{GlobalSetup}}(1^\lambda)$.

Every time $\mathcal{O}_{\mathsf{create}}$ is queried, it generates $(\Gamma.\overline{\mathsf{pk}}^1, \Gamma.\overline{\mathsf{msk}}^1), \ldots, (\Gamma.\overline{\mathsf{pk}}^B, \Gamma.\overline{\mathsf{msk}}^B)$ by running $\Gamma.\overline{\mathsf{Setup}}(\Gamma.\overline{\mathsf{gp}})$, and it computes the key pair $(\mathsf{pk}, \mathsf{sk})$ for the newly created authority, where $\mathsf{pk} = (\Gamma.\overline{\mathsf{pk}}^1, \ldots, \Gamma.\overline{\mathsf{pk}}_B)$, and $\mathsf{sk} = (\Gamma.\overline{\mathsf{msk}}_1, \ldots, \Gamma.\overline{\mathsf{msk}}_B)$.

When $\mathcal{A}$ sends $(\boldsymbol{M}, \rho, \Pi)$, the challenge ciphertext $\mathsf{ct}^\star$ is computed as follows: $\mathsf{ct}^\star = \{\Gamma.\mathsf{ct}_j\}_{j\in[\ell]}$, where for all $j \in [\ell]$, if $\theta(j)$ is an honest authority with key pair $(\mathsf{pk}_{\theta(j)}, \mathsf{sk}_{\theta(j)})$ where $\mathsf{pk}_{\theta(j)} = (\Gamma.\overline{\mathsf{pk}}^1_{\theta(j)}, \ldots, \Gamma.\overline{\mathsf{pk}}^B_{\theta(j)})$, we have: $\Gamma.\mathsf{ct}_j \leftarrow \Gamma.\overline{\mathsf{Enc}}(\Gamma.\mathsf{td}, \Gamma.\overline{\mathsf{pk}}^{\xi(j)}_{\theta(j)}, \rho(j), \mathit{leakage})$, where $\mathit{leakage} = \emptyset$. Note that by the decomposable property of Γ, $\Gamma.\mathsf{ct}_j = (\mathsf{hd}, \mathsf{pl})$, where the header hd is the same for all $\Gamma.\mathsf{ct}_j$. If $\theta(j)$ is a corrupted authority, then $\Gamma.\mathsf{ct}_j = (\mathsf{hd}, \mathsf{pl})$, where hd is the same as the header for honest authorities (we know there is at least one honest authority in the set Π since $\mathcal{A}$ is admissible), and $\mathsf{pl} = \Gamma.\mathsf{Enc}_{\mathsf{pl}}\big(\Gamma.\mathsf{msk}^{\xi(j)}_{\theta(j)}, \rho(j), (s_j, u_j), \mathsf{hd}\big)$, where secret keys $\Gamma.\mathsf{msk}^1_{\theta(j)}, \ldots, \Gamma.\mathsf{msk}^B_{\theta(j)}$ are provided as in the previous game.

Later on, when $\mathcal{A}$ queries its oracle $\mathcal{O}_{\mathsf{create}}$, a new honest authority is created in the same way as before. When $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{KeyGen}}$ on input $(\mathsf{a}, \mathsf{gid}, \mathsf{pk})$ where pk is an honest authority, the game returns $\mathsf{sk}_{\mathsf{gid},\mathsf{a}} = (\Gamma.\overline{\mathsf{sk}}^1, \ldots, \Gamma.\overline{\mathsf{sk}}^B)$ where for all $i \in [B]$, $\Gamma.\overline{\mathsf{sk}}^i \leftarrow \Gamma.\overline{\mathsf{KeyGen}}(\Gamma.\mathsf{td}, \mathsf{a}, \mathsf{pk}, \mathit{leakage})$, where $\mathit{leakage}$ is the set of all pairs $(\llbracket s_j + z_{\mathsf{gid}'} u_j \rrbracket_2, \rho(j))$ for all queries $(\rho(j), \theta(j), \mathsf{gid}')$ sent by $\mathcal{A}$ to $\mathcal{O}_{\mathsf{KeyGen}}$.

Here, $(\overline{\mathsf{GlobalSetup}}, \overline{\mathsf{Setup}}, \overline{\mathsf{Enc}}, \overline{\mathsf{KeyGen}})$ is a simulator for Γ, which is a selective multi-instance secure ID-based inner-product FE. It is clear that Game_1 corresponds to the real experiment for the security of Γ, whereas Game_2 corresponds to the ideal experiment. Thus, we have $\mathsf{Game}_1 \approx_c \mathsf{Game}_2$.

$\underline{\mathsf{Game}_3}$: is the same as Game_2 except for the shares $s_1, \ldots, s_\ell$ used to generate the challenge ciphertext and the functional secret keys. In Game_2, they are generated as $s_j = (s, \boldsymbol{w})^\top \boldsymbol{M}_j$ for all $j \in [\ell]$, where $\boldsymbol{M}_j$ denotes the j'th column of $\boldsymbol{M}$, and $\boldsymbol{w} \leftarrow_R \mathbb{Z}_p^{n-1}$. In Game_3 however, the shares are generated as $s_j = (s, \boldsymbol{w})^\top \boldsymbol{M}_j + s(0, \boldsymbol{w}^\star)^\top \boldsymbol{M}_j$ where $\boldsymbol{w} \leftarrow_R \mathbb{Z}_p^{\ell-1}$ as before, and $(1, \boldsymbol{w}^\star) \in \mathbb{Z}_p^n$ is a vector sampled uniformly at random in the affine space of vectors whose first coordinate is 1 and which are orthogonal to all vectors $\{\boldsymbol{M}_j\}_{j:\theta(j)\in\mathcal{S}_{\mathsf{corr}}}$, where $\mathcal{S}_{\mathsf{corr}}$ denotes the set of corrupted authorities. We call $A^\perp_{\mathcal{S}_{\mathsf{corr}}}$ this affine space. Since $\mathcal{A}$ is admissible, we know that $\{\boldsymbol{M}_j\}_{j:\theta(j)\in\mathcal{S}_{\mathsf{corr}}}$ does not span the vector $\mathbf{1} \in \mathbb{Z}_p^n$, so $A^\perp_{\mathcal{S}_{\mathsf{corr}}}$ is not empty. The fact that Game_2 and Game_3 are identically distributed follows from the fact that for all $\boldsymbol{w}^\star \in \mathbb{Z}_p^{n-1}$ and $s \in \mathbb{Z}_p$, the following distributions are identical: $\{\boldsymbol{w} \leftarrow_R \mathbb{Z}_p^{n-1} : \boldsymbol{w}\}$ and $\{\boldsymbol{w} \leftarrow_R \mathbb{Z}_p^{n-1} : \boldsymbol{w} + s\boldsymbol{w}^\star\}$. The first distribution corresponds to Game_2 (with some post-processing), whereas the second distribution corresponds to Game_3 (with the same post-processing). Note that the challenge ciphertext in Game_3 does not depend on s anymore, thanks to the orthogonality properties of $(1, \boldsymbol{w}^\star)$. Namely, for any $j \in [\ell]$ such that $\theta(j)$ is corrupted, we have $s_j = (s, \boldsymbol{w})^\top \boldsymbol{M}_j + s(0, \boldsymbol{w}^\star)^\top \boldsymbol{M}_j = (0, \boldsymbol{w})^\top \boldsymbol{M}_j + s(1, \boldsymbol{w}^\star)^\top \boldsymbol{M}_j = (0, \boldsymbol{w})^\top \boldsymbol{M}_j$. It only appears in $\mathit{leakage}$ used to generate the functional secret keys as $\llbracket (s_j + z_{\mathsf{gid}} u_j \rrbracket_2 = \llbracket (0, \boldsymbol{w})^\top \boldsymbol{M}_j + s(1, \boldsymbol{w}^\star)^\top \boldsymbol{M}_j + z_{\mathsf{gid}} u_j \rrbracket_2$ for all queries $(\rho(j), \theta(j), \mathsf{gid})$ sent by $\mathcal{A}$ to $\mathcal{O}_{\mathsf{KeyGen}}$.

$\underline{\mathsf{Game}_4}$: is the same as Game_3 except for the shares $u_1, \ldots, u_\ell$ used to generate the challenge ciphertext and the functional secret keys. In Game_3, they are generated as $u_j = (0, \boldsymbol{h})^\top \boldsymbol{M}_j$ for all $j \in [\ell]$, where $\boldsymbol{h} \leftarrow_R \mathbb{Z}_p^{n-1}$. In Game_4 however, they are generated as $u_j = (0, \boldsymbol{h})^\top \boldsymbol{M}_j + (0, \boldsymbol{u}^\star)^\top \boldsymbol{M}_j$ where $\boldsymbol{h} \leftarrow_R \mathbb{Z}_p^{n-1}$ as before and $(1, \boldsymbol{u}^\star)^\top \in \mathbb{Z}_p^\ell$ is a vector sampled uniformly at random in $A^\perp_{\mathcal{S}_{\mathsf{corr}}}$. Game_3 and Game_4 are

identically distributed, which follows from the fact that for all $\boldsymbol{u}^\star \in \mathbb{Z}_p^{n-1}$, the following distributions are identical: $\{\boldsymbol{h} \leftarrow_R \mathbb{Z}_p^{n-1} : \boldsymbol{h}\}$ and $\{\boldsymbol{h} \leftarrow_R \mathbb{Z}_p^{n-1} : \boldsymbol{h} + \boldsymbol{u}^\star\}$. The first distribution corresponds to Game_3 (with some post-processing), whereas the second distribution corresponds to Game_4 (with the same post-processing). Note that the vector $\boldsymbol{u}^\star$ does not appear in the challenge ciphertext in Game_4, thanks to the orthogonality properties of $(1, \boldsymbol{u}^\star)$. Namely, for any $j \in [\ell]$ such that $\theta(j)$ is corrupted, we have $u_j = (0, \boldsymbol{h})^\top \boldsymbol{M}_j + (0, \boldsymbol{u}^\star)^\top \boldsymbol{M}_j = (-1, \boldsymbol{h})^\top \boldsymbol{M}_j + (1, \boldsymbol{u}^\star)^\top \boldsymbol{M}_j = (-1, \boldsymbol{h})^\top \boldsymbol{M}_j$. It only appears in *leakage* used to generate the functional secret keys as $[\![s_j + z_{\mathsf{gid}} u_j]\!]_2 = [\![s_j + z_{\mathsf{gid}}(-1, \boldsymbol{h})^\top \boldsymbol{M}_j + z_{\mathsf{gid}}(1, \boldsymbol{u}^\star)^\top \boldsymbol{M}_j]\!]_2$ for all queries $(\rho(j), \theta(j), \mathsf{gid})$ sent by $\mathcal{A}$ to $\mathcal{O}_{\mathsf{KeyGen}}$.

Game_5: is the same as Game_4 except that *leakage* which is used to generate the functional secret keys is now defined as the set of all pairs: $([\![s_j + z_{\mathsf{gid}} u_j + \left(\boldsymbol{M}^\perp_{\mathcal{S}_{\mathsf{corr}}} \boldsymbol{v}_{\mathsf{gid}}\right)^\top \boldsymbol{M}_j]\!]_2, \rho(j))$ for all queries $(\rho(j), \theta(j), \mathsf{gid})$ sent by $\mathcal{A}$ to $\mathcal{O}_{\mathsf{KeyGen}}$, where $\boldsymbol{M}^\perp_{\mathcal{S}_{\mathsf{corr}}} \in \mathbb{Z}_p^{\ell \times d}$ is a basis for the vector space that contains all vectors orthogonal to $\{\boldsymbol{M}_j\}_{j:\theta(j) \in \mathcal{S}_{\mathsf{corr}}}$, which is of dimension d (d may be 0), and for all gid, $\boldsymbol{v}_{\mathsf{gid}} \leftarrow_R \mathbb{Z}_p^d$.

We prove that $\mathsf{Game}_4 \approx_c \mathsf{Game}_5$ using the DDH assumption in $\mathbb{G}_2$, in the random oracle model. First, recall that for all $j \in [\ell]$, we have:

$$[\![s_j + z_{\mathsf{gid}} u_j]\!]_2 = [\![(0, \boldsymbol{w})^\top \boldsymbol{M}_j + s(1, \boldsymbol{w}^\star)^\top \boldsymbol{M}_j + z_{\mathsf{gid}}(-1, \boldsymbol{h})^\top \boldsymbol{M}_j + z_{\mathsf{gid}}(1, \boldsymbol{u}^\star)^\top \boldsymbol{M}_j]\!]_2 ,$$

where $(1, \boldsymbol{w}^\star)$ and $(1, \boldsymbol{u}^\star)$ are uniformly random over $A^\perp_{\mathcal{S}_{\mathsf{corr}}}$. For any vector $\boldsymbol{x} \in A^\perp_{\mathcal{S}_{\mathsf{corr}}}$, we can write $A^\perp_{\mathcal{S}_{\mathsf{corr}}} = \boldsymbol{x} + \mathsf{Span}(\boldsymbol{M}^\perp_{\mathcal{S}_{\mathsf{corr}}})$. Thus, we can write $(1, \boldsymbol{u}^\star) = (1, \boldsymbol{w}^\star) + \boldsymbol{M}^\perp_{\mathcal{S}_{\mathsf{corr}}} \boldsymbol{v}$ where $\boldsymbol{v} \leftarrow_R \mathbb{Z}_p^d$.

Then, we argue that for all gid, $([\![z_{\mathsf{gid}}]\!]_2, [\![z_{\mathsf{gid}} \boldsymbol{v}]\!]_2) \approx_c ([\![z_{\mathsf{gid}}]\!]_2, [\![\boldsymbol{v}_{\mathsf{gid}}]\!]_2) \equiv ([\![z_{\mathsf{gid}}]\!]_2, [\![z_{\mathsf{gid}} \boldsymbol{v} + \boldsymbol{v}_{\mathsf{gid}}]\!]_2)$, where $\boldsymbol{v}_{\mathsf{gid}} \leftarrow_R \mathbb{Z}_p^d$. Note that the first distribution corresponds to Game_4, whereas the last distribution corresponds to Game_5. The computational indistinguishability (denoted by $\approx_c$) is justified by the DDH assumption in $\mathbb{G}_2$, whereas the second equality is information theoretic. Note that in fact to use the DDH assumption, we first need to argue that the values $[\![z_{\mathsf{gid}}]\!]_2$ can be generated as truly random values computed on the fly. Then, we need to apply a hybrid argument over all gid that are part of a query to $\mathcal{O}_{\mathsf{KeyGen}}$, and guess which query to the random oracle is going to correspond to the gid of the current hybrid, so that the DDH challenge can be embedded in the output of $H(\mathsf{gid})$. We refrain from giving too many details on this argument which is routine in proofs in the ROM.

Game_6: is the same as Game_5, except that *leakage* is now the set of all pairs of the form: $([\![s_j + z_{\mathsf{gid}} u_j + \left(\boldsymbol{M}^\perp_{\mathcal{S}_{\mathsf{corr}}} s \cdot \boldsymbol{v}_{\mathsf{gid}}\right)^\top \boldsymbol{M}_j]\!]_2, \rho(j))$, for all queries $(\rho(j), \theta(j), \mathsf{gid})$ sent by $\mathcal{A}$ to $\mathcal{O}_{\mathsf{KeyGen}}$. We have $\mathsf{Game}_5 \approx_s \mathsf{Game}_6$, which follows from the fact that the following distributions have statistical distance $1/p$: $\{s \leftarrow_R \mathbb{Z}_p, \forall \mathsf{gid} \in \mathcal{Q}, \boldsymbol{v}_{\mathsf{gid}} \leftarrow_R \mathbb{Z}_p^d : \left(s, \{\boldsymbol{v}_{\mathsf{gid}}\}_{\mathsf{gid} \in \mathcal{Q}}\right)\}$ and $\{s \leftarrow_R \mathbb{Z}_p, \forall \mathsf{gid} \in \mathcal{Q}, \boldsymbol{v}_{\mathsf{gid}} \leftarrow_R \mathbb{Z}_p^d : \left(s, \{s \cdot \boldsymbol{v}_{\mathsf{gid}}\}_{\mathsf{gid} \in \mathcal{Q}}\right)\}$, where $\mathcal{Q}$ denotes the set of queried gid. Note that the first distribution corresponds to Game_5 (with some post processing), whereas the second distribution corresponds to Game_6 (with the same post-processing).

Game_7: is the same as Game_6, except that *leakage* is now the set of all pairs of the form: $([\![(0, \boldsymbol{w})^\top \boldsymbol{M}_j + u_j + \left(\boldsymbol{M}^\perp_{\mathcal{S}_{\mathsf{corr}}} s \cdot \boldsymbol{v}_{\mathsf{gid}}\right)^\top \boldsymbol{M}_j]\!]_2, \rho(j))$, for all queries $(\rho(j), \theta(j), \mathsf{gid})$ sent by $\mathcal{A}$ to $\mathcal{O}_{\mathsf{KeyGen}}$. We show that Game_7 is identically distributed to Game_6 as follows.

For all gid that is part of a query to $\mathcal{O}_{\mathsf{KeyGen}}$, we define $\mathcal{S}_{\mathsf{gid}} = \cup_{\mathsf{pk} \in \mathcal{S}_{\mathsf{corr}}} \mathcal{U}_{\mathsf{pk}} \cup \{\mathsf{a} \text{ s.t. } \exists \mathsf{pk} \in \mathcal{S}_{\mathsf{hon}}, (\mathsf{pk}, \mathsf{gid}, \mathsf{a}) \in \mathcal{Q}_{\mathsf{KeyGen}}\}$, where $\mathcal{S}_{\mathsf{hon}}$ denotes the set of honest authorities, and $\mathcal{Q}_{\mathsf{KeyGen}}$ denotes the set of queries made to $\mathcal{O}_{\mathsf{KeyGen}}$; and we define the affine space $A^\perp_{\mathcal{S}_{\mathsf{gid}}}$ as the set of all vectors whose first coordinate is 1 and which are orthogonal to $\{\boldsymbol{M}_j\}_{j:\rho(j) \in \mathcal{S}_{\mathsf{gid}}}$. The adversary $\mathcal{A}$ is admissible, which exactly means that for all gid that are part of a query to $\mathcal{Q}_{\mathsf{KeyGen}}$, $\mathcal{S}_{\mathsf{gid}}$ does not satisfies the access structure $(\boldsymbol{M}, \rho)$, i.e. $A^\perp_{\mathcal{S}_{\mathsf{gid}}}$ is not empty, and let $(1, \boldsymbol{w}^\star_{\mathsf{gid}}) \in A^\perp_{\mathcal{S}_{\mathsf{gid}}}$. It is clear that $(1, \boldsymbol{w}^\star_{\mathsf{gid}}) \in A^\perp_{\mathcal{S}_{\mathsf{corr}}}$. Thus, for all gid that are part of a query to $\mathcal{Q}_{\mathsf{KeyGen}}$ we can write $A^\perp_{\mathcal{S}_{\mathsf{corr}}} = (1, \boldsymbol{w}^\star_{\mathsf{gid}}) + \mathsf{Span}(\boldsymbol{M}^\perp_{\mathcal{S}_{\mathsf{corr}}})$.

For all $j \in [\ell]$ and gid such that $(\rho(j), \theta(j), \mathsf{gid}) \in \mathcal{Q}_{\mathsf{KeyGen}}$, we have:

$$\begin{aligned}
s_j + \left(\boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp} s \cdot \boldsymbol{v}_{\mathsf{gid}}\right)^{\top} \boldsymbol{M}_j &= (0, \boldsymbol{w})^{\top} \boldsymbol{M}_j + s(1, \boldsymbol{w}^{\star})^{\top} \boldsymbol{M}_j + \boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp} s \cdot \boldsymbol{v}_{\mathsf{gid}} \\
&= (0, \boldsymbol{w})^{\top} \boldsymbol{M}_j + s \cdot \underbrace{\left((1, \boldsymbol{w}^{\star}) + \boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp} \boldsymbol{v}_{\mathsf{gid}}\right)^{\top}}_{\text{uniformly random from } A_{\mathcal{S}_{\mathsf{corr}}}^{\perp}} \boldsymbol{M}_j \\
&= (0, \boldsymbol{w})^{\top} \boldsymbol{M}_j + s \cdot \underbrace{\left((1, \boldsymbol{w}_{\mathsf{gid}}^{\star}) + \boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp} \boldsymbol{v}_{\mathsf{gid}}\right)^{\top}}_{\text{uniformly random from } A_{\mathcal{S}_{\mathsf{corr}}}^{\perp}} \boldsymbol{M}_j \\
&= (0, \boldsymbol{w})^{\top} \boldsymbol{M}_j + s \cdot \left(\boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp} \boldsymbol{v}_{\mathsf{gid}}\right)^{\top} \boldsymbol{M}_j + \underbrace{(1, \boldsymbol{w}_{\mathsf{gid}}^{\star})^{\top} \boldsymbol{M}_j}_{=0} \\
&= (0, \boldsymbol{w})^{\top} \boldsymbol{M}_j + \left(\boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp} s \cdot \boldsymbol{v}_{\mathsf{gid}}\right)^{\top} \boldsymbol{M}_j.
\end{aligned}$$

In the second equality, we use the fact that the value $\left((1, \boldsymbol{w}^{\star}) + \boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp} \boldsymbol{v}_{\mathsf{gid}}\right)^{\top} \boldsymbol{M}_j$ is uniformly random $A_{\mathcal{S}_{\mathsf{corr}}}^{\perp}$. This is because $(1, \boldsymbol{w}^{\star}) \in A_{\mathcal{S}_{\mathsf{corr}}}^{\perp}$, so we can write $A_{\mathcal{S}_{\mathsf{corr}}}^{\perp} = (1, \boldsymbol{w}^{\star}) + \mathsf{Span}(\boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp})$. In the penultimate equality, we used the fact that for all $j \in [\ell]$ such that $(\rho(j), \theta(j), \mathsf{gid}) \in \mathcal{Q}_{\mathsf{KeyGen}}$, we have $(1, \boldsymbol{w}_{\mathsf{gid}}^{\star})^{\top} \boldsymbol{M}_j$. This is because $(1, \boldsymbol{w}_{\mathsf{gid}}^{\star}) \in A_{\mathcal{S}_{\mathsf{corr}}}^{\perp}$. The first distribution corresponds to Game_6 (with some post-processing), whereas the last distribution corresponds to Game_7 (with the same post-processing). Thus, we have $\mathsf{Game}_6 = \mathsf{Game}_7$.

$\underline{\mathsf{Game}_8}$: is the same as Game_7, except that *leakage* is now the set of all pairs of the form: $(\llbracket (0, \boldsymbol{w}^{\top}) \boldsymbol{M}_j + u_j + \left(\boldsymbol{M}_{\mathcal{S}_{\mathsf{corr}}}^{\perp} \boldsymbol{v}_{\mathsf{gid}}\right)^{\top} \boldsymbol{M}_j \rrbracket_2, \rho(j))$, for all queries $(\rho(j), \theta(j), \mathsf{gid}) \in \mathcal{Q}_{\mathsf{KeyGen}}$. This is the reverse transition that from Game_5 to Game_6 (where we switch from $\boldsymbol{v}_{\mathsf{gid}}$ to $s \cdot \boldsymbol{v}_{\mathsf{gid}}$), so we have $\mathsf{Game}_7 \approx_s \mathsf{Game}_8$, following the same statistical argument. Note that in Game_8 the view of $\mathcal{A}$ does not depend on the secret $s \leftarrow_R \mathbb{Z}_p$ that is used to compute $\kappa_0 = \llbracket s \rrbracket_{\mathsf{t}}$. Thus, the advantage of $\mathcal{A}$ in Game_8 is 0, which conclude the proof. □

References

ABGW17. Miguel Ambrona, Gilles Barthe, Romain Gay, and Hoeteck Wee. Attribute-based encryption in the generic group model: Automated proofs and new constructions. In *ACM CCS 17*, pages 647–664. ACM Press, 2017. 23, 24

ABS16. Miguel Ambrona, Gilles Barthe, and Benedikt Schmidt. Automated unbounded analysis of cryptographic constructions in the generic group model. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 822–851. Springer, Heidelberg, May 2016. 23

AC16. Shashank Agrawal and Melissa Chase. A study of pair encodings: Predicate encryption in prime order groups. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 259–288. Springer, Heidelberg, January 2016. 2

ACGU20. Michel Abdalla, Dario Catalano, Romain Gay, and Bogdan Ursu. Inner-product functional encryption with fine-grained access control. LNCS, pages 467–497. Springer, Heidelberg, December 2020. 4, 6, 8

AGHO11. Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 649–666. Springer, Heidelberg, August 2011. 23

AGOT14. Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Structure-preserving signatures from type II pairings. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 390–407. Springer, Heidelberg, August 2014. 23

AGT20. Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-party functional encryption. Cryptology ePrint Archive, Report 2020/1266, 2020. https://eprint.iacr.org/2020/1266. 4

AKOT15. Masayuki Abe, Markulf Kohlweiss, Miyako Ohkubo, and Mehdi Tibouchi. Fully structure-preserving signatures and shrinking commitments. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 35–65. Springer, Heidelberg, April 2015. 23

Att16. Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Heidelberg, December 2016. 2

BBG05. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, Heidelberg, May 2005. 23

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Heidelberg, August 2004. 23

Bei96. Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. Ph.D., Technion - Israel Institute of Technology, 1996. 5

BFF+14. Gilles Barthe, Edvard Fagerholm, Dario Fiore, John C. Mitchell, Andre Scedrov, and Benedikt Schmidt. Automated analysis of cryptographic assumptions in generic group models. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 95–112. Springer, Heidelberg, August 2014. 23

BFF+15. Gilles Barthe, Edvard Fagerholm, Dario Fiore, Andre Scedrov, Benedikt Schmidt, and Mehdi Tibouchi. Strongly-optimal structure preserving signatures from type II pairings: Synthesis and lower bounds. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 355–376. Springer, Heidelberg, March / April 2015. 23

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988. 24

CC09. Melissa Chase and Sherman S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM CCS 09*, pages 121–130. ACM Press, November 2009. 2

CGW15. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, April 2015. 2

Cha07. Melissa Chase. Multi-authority attribute based encryption. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 515–534. Springer, Heidelberg, February 2007. 1

Cra97. Ronald Cramer. Modular design of secure yet practical cryptographic protocols. 1997. 25

DKW21. Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority abe for dnfs from lwe. In *Eurocrypt*, 2021. 4

DP19. Edouard Dufour Sans and David Pointcheval. Unbounded inner-product functional encryption with succinct keys. In *ACNS 19*, LNCS, pages 426–441. Springer, Heidelberg, 2019. 4, 6

Fre10. David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 44–61. Springer, Heidelberg, May 2010. 2

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. 25

GK15. Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 253–280. Springer, Heidelberg, April 2015. 25

GMR85. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985. 24

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309. 1

Gro15. Jens Groth. Efficient fully structure-preserving signatures for large messages. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 239–259. Springer, Heidelberg, November / December 2015. 23

Gui13. Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS 13*, volume 7954 of *LNCS*, pages 357–372. Springer, Heidelberg, June 2013. 2

Ins18. European Telecommunications Standards Institute. Etsi releases cryptographic standards for secure access control, 2018. https://www.etsi.org/newsroom/press-releases/1328-2018-08-press-etsi-releases-cryptographic-standards-for-secure-access-control. 2

JS08. Tibor Jager and Jörg Schwenk. On the equivalence of generic group models. In Joonsang Baek, Feng Bao, Kefei Chen, and Xuejia Lai, editors, *ProvSec 2008*, volume 5324 of *LNCS*, pages 200–209. Springer, Heidelberg, October / November 2008. 23

Kim19. Sam Kim. Multi-authority attribute-based encryption from lwe in the ot model. *IACR Cryptol. ePrint Arch.*, 2019:280, 2019. 4

KLM+18. Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J. Wu. Function-hiding inner product encryption is practical. In *SCN 18*, LNCS, pages 544–562. Springer, Heidelberg, 2018. 23

KW93. M. Karchmer and A. Wigderson. On span programs. In *Structure in Complexity Theory Conference, 1993., Proceedings of the Eighth Annual*, pages 102–111, May 1993. 5

LCLS08. Huang Lin, Zhenfu Cao, Xiaohui Liang, and Jun Shao. Secure threshold multi authority attribute based encryption without a central authority. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT 2008*, volume 5365 of *LNCS*, pages 426–436. Springer, Heidelberg, December 2008. 2

Lew12. Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 318–335. Springer, Heidelberg, April 2012. 2

LW11. Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 568–588. Springer, Heidelberg, May 2011. 2, 3, 9

Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005. 23

MJ18. Yan Michalevsky and Marc Joye. Decentralized policy-hiding ABE with receiver privacy. In *ESORICS 2018, Part II*, LNCS, pages 548–567. Springer, Heidelberg, September 2018. 4

MKE08. Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. Distributed attribute-based encryption. In *International Conference on Information Security and Cryptology*, pages 20–36. Springer, 2008. 1

Nec94. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994. 22, 23

OT09. Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical predicate encryption for inner-products. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 214–231. Springer, Heidelberg, December 2009. 2

Rog15. Phillip Rogaway. The moral character of cryptographic work. *IACR Cryptol. ePrint Arch.*, 2015:1162, 2015. 1

Sch80. J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980. 23

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997. 23

SJ00. Claus-Peter Schnorr and Markus Jakobsson. Security of signed ElGamal encryption. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 73–89. Springer, Heidelberg, December 2000. 23

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005. 1

TT18. Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. In *ASIACRYPT 2018, Part II*, LNCS, pages 609–639. Springer, Heidelberg, December 2018. 4

Zip79. Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation*, pages 216–226, Berlin, Heidelberg, 1979. Springer Berlin Heidelberg. 23

Supplementary Material

A The Generic Group Model

The generic group model is an idealized cryptographic model, first introduced by Nechaev [Nec94], designed to analyze the security of cryptographic constructions (defined over an algebraic group) with respect to adversaries that do not exploit the representation of the group (and, consequently, are called *generic*).

This model can be used to argue that (possibly non-standard) cryptographic assumptions meet minimal requirements of security. Furthermore, although we do not advocate proving security of cryptographic constructions in the generic group model over the standard model, generic security can be considered sufficient for practical use, when the primitives are implemented over group representations for which all best known algorithms are generic.

Shoup, Schnorr and Jakobsson, were among the first to prove security of cryptographic constructions in the generic group model. In particular, Shoup [Sho97] proved generic security of an identification scheme, whereas Schnorr and Jakobsson [SJ00] focused on signed ElGamal encryption. The model became an important tool for establishing the security of pairing-based cryptographic constructions [Gro15, AKOT15, AGHO11, AGOT14, BFF+15, KLM+18] and even several automated tools have been developed to prove security in this model [BFF+14, BFF+15, ABS16, ABGW17] given that the generic group model is particularly amenable for symbolic reasoning.

Generic adversaries have been modeled in different and independent ways in the literature, for example, Nechaev [Nec94], Shoup [Sho97] and Maurer [Mau05] provided independent (although equivalent [JS08]) approaches, which were then extended to the framework of bilinear groups [BBS04, BBG05]. The approach of Nechaev and Shoup lets the adversary access group elements through a randomly selected representation. Instead, we adopt Maurer's approach, which gives the adversary oracle access to the group elements via so-called *handles* or identifiers. The adversary can evaluate the group law between group elements, by referring to their handles, and will be given a fresh handle pointing to the result of the evaluation. (Other operations like pairing evaluation can be modeled in a similar fashion, and the adversary is allowed to perform equality checks between the content of their available handles.) Note that they way the group is internally implemented is completely transparent to the adversary. In fact, any group of order n could be simply internally implemented as $(\mathbb{Z}_n, +)$.

The symbolic group model. The symbolic group model is a very useful abstraction that allows us to reduce complex security experiments to the satisfiability of purely algebraic conditions. In a nutshell, an experiment in the symbolic group model is exactly as our above generic group model experiment, where the underlying group that the adversary is given access to (via handles) is implemented by formal polynomials. More precisely, a group of order n is represented by polynomials with coefficients over $\mathbb{Z}_n$, involving formal variables that correspond to randomly sampled values during the experiment. (Note that equality checks, in this case, are performed between polynomials.) Also, note that such a representation makes the experiment completely deterministic, an essential property that can be used to argue that if a certain system of polynomial equations is unsatisfiable, then all adversaries have 0 probability of winning the symbolic experiment.

Once a cryptographic construction is shown to be symbolically secure, its security can be extended to the desired generic group model by using the fact that any adversary, performing a polynomial number of operations, has a negligible advantage in distinguishing between the symbolic and the generic implementations of their underlying group. This step is usually proven once in a so-called *master theorem* [BBG05, BFF+14, ABS16, ABGW17], usually giving an explicit bound on the distinguishing probability, dependent on the degree of the polynomials that the adversary can create after polynomially many queries to their oracles. In a nutshell, these master theorems exploit the fact that the only way that an adversary can distinguish between the symbolic and the generic implementation, is by triggering a so-called *bad event*. Namely, a successful equality check that would not hold in the symbolic world (which is deterministic and can therefore be simulated by the adversary). The probability of such bad events is commonly bounded by the Schwartz-Zippel lemma [Sch80, Zip79]:

Lemma 1 (Schwartz-Zippel). *Let K be a field, let $f \in K[X_1, \ldots, X_n]$ be a non-zero polynomial of degree d and let S be a finite subset of K. It holds:*

$$\Pr\left[\, r_1, \ldots, r_n \leftarrow_R S \,:\, f(r_1, \ldots, r_n) = 0 \,\right] \;\leq\; d/|S| \;\; .$$

Roughly, the lemma guarantees that any possible equality check between different polynomials (which does not hold in the symbolic model) will hold in the generic model only with negligible probability (when

the value of the formal variables is chosen uniformly at random in a set S that is much larger than the degree d). Extra care needs to be taken into account when the adversary performs queries adaptively, since future queries could potentially depend on already sampled values, but standard hybrid arguments over the adversary's queries can be used to formalize these cases (for example, see [ABGW17]).

Given that proofs in the generic group model have been intensively studied in the literature and the relations between symbolic security and actual generic security are now well consolidated, in this work we will focus on proving the symbolic security of our primitives. Our symbolic model includes a formal variable for every group element (uniformly) sampled during the corresponding security experiment. Furthermore, in order to model our random oracle that produces group elements, we consider a formal variable for every possible random oracle input.

B Zero-knowledge proofs

A zero-knowledge (ZK) proof [GMR85] is a two party protocol executed between a prover and a verifier that allows the prover to convince the verifier about the validity of a certain statement, without revealing any other information, e.g., why the statement is true. More formally, given a binary relation $R : \mathcal{X} \times \mathcal{W} \to \{0,1\}$ defined over a set of statements $\mathcal{X}$ and a set of witnesses $\mathcal{W}$, let L_R be the language defined as $L_R \coloneqq \{x \in \mathcal{X} \mid \exists w \in \mathcal{W} : R(x,w) = 1\}$, a zero-knowledge proof system allows a prover in possession of $(x,w) \in R$ to convince a verifier of the fact that $x \in L_R$ without revealing any information about w. The soundness property ensures that no proof can convince the verifier of the validity of a false statement. Non-interactive ZK proof systems [BFM88] are a version of ZK proof systems where the prover sends one single message to the verifier, no further interaction is required. Proof systems that only satisfy computational soundness are called *argument systems*. We require a strong soundness property which coins the system an *argument of knowledge* defined below.

Definition 4 (NIZK-AoK). *A* Non-Interactive Zero-Knowledge Argument of Knowledge (NIZK-AoK) *for a relation R consists of the following PPT algorithms:*

- $\mathsf{CRSGen}(1^\lambda) \to \mathsf{crs}$. *On input the security parameter, it (probabilistically) generates a common reference string.*
- $\mathsf{Prove}(\mathsf{crs}, x, w) \to \pi$. *On input a* crs*, a statement x and a witness w, it (probabilistically) generates an argument π.*
- $\mathsf{Verify}(\mathsf{crs}, x, \pi) \to 1/0$. *On input the* crs*, a statement and an argument, it deterministically outputs a bit representing acceptance (1) or rejection (0).*

Completeness. For all $(x,w) \in R$, all $\lambda \in \mathbb{N}$, all crs in the support of $\mathsf{CRSGen}(1^\lambda)$, and all π in the support of $\mathsf{Prove}(\mathsf{crs}, x, w)$, it holds that $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1$.

Zero-knowledge. There exist two additional PPT algorithms $\overline{\mathsf{CRSGen}}$ and $\overline{\mathsf{Prove}}$ such that for every PPT adversary $\mathcal{A}$, the following difference is negligible in λ:

$$\Pr\left[\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda) : 1 = \mathcal{A}^{\mathcal{O}_\mathsf{r}}(\mathsf{crs})\right] - \Pr\left[(\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \overline{\mathsf{CRSGen}}(1^\lambda) : 1 = \mathcal{A}^{\mathcal{O}_\mathsf{s}}(\mathsf{crs})\right] ,$$

where $\mathcal{O}_\mathsf{r}$, $\mathcal{O}_\mathsf{s}$ are oracles that take (x,w) as input, and return $\mathsf{Prove}(\mathsf{crs}, x, w)$ and $\overline{\mathsf{Prove}}(\overline{\mathsf{crs}}, \mathsf{td}, x)$ respectively if $R(x,w) = 1$; they both return $\perp$ otherwise.

Knowledge-soundness. There exists a polynomial-time *extractor* $\mathcal{E}$ such that for all PPT adversaries $\mathcal{A}$, the following probability is negligible in λ:

$$\Pr\left[\begin{array}{c} (\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \overline{\mathsf{CRSGen}}(1^\lambda) \\ (x,\pi) \leftarrow \mathcal{A}^{\overline{\mathsf{Prove}}(\overline{\mathsf{crs}}, \mathsf{td}, \cdot)}(\overline{\mathsf{crs}}) \\ w \leftarrow \mathcal{E}^{\mathcal{A}}(\overline{\mathsf{crs}}, x, \pi) \end{array} : \mathsf{Verify}(\overline{\mathsf{crs}}, x, \pi) = 1 \ \wedge \ R(x,w) = 0\right] ,$$

where the adversary is restricted to not have queried x to its oracle.

Σ-protocols, introduced in [Cra97], are public-coin interactive protocols that consist of only three data transfers between the prover and the verifier, and that must satisfy weaker notions of zero-knowledge and so-called special soundness. We refer to [GK15] for a formal definition of Σ-protocols and note that they can be compiled into fully secure NIZK systems by leveraging the Fiat-Shamir heuristic [FS87].

C Security Proof of the ID-based Inner-Product FE from SXDH

Proof (of Theorem 1). We proceed via a series of hybrid games described bellow (the differences from one game to the next are highlighted in red).

Game_0: is the real experiment from the security definition in Section 3.1. We recall it here for completeness. The adversary $\mathcal{A}$ first receives $\mathsf{gp} = (\mathcal{PG}, [\![\boldsymbol{a}]\!]_1)$. Then, it can query its oracle $\mathcal{O}_{\mathsf{create}}$, that on the i-th query, creates the new pair $(\mathsf{pk}_i, \mathsf{msk}_i)$ where $\mathsf{pk}_i = ([\![\boldsymbol{V}^i\boldsymbol{a}]\!]_1, [\![\boldsymbol{U}_0^i\boldsymbol{a}]\!]_1, [\![\boldsymbol{U}_1^i\boldsymbol{a}]\!]_1)$ and $\mathsf{msk}_i = (\boldsymbol{V}^i, \boldsymbol{U}_0^i, \boldsymbol{U}_1^i)$ where $\boldsymbol{V}^i \leftarrow_R \mathbb{Z}_p^{d\times 2}$, $\boldsymbol{U}_0^i, \boldsymbol{U}_1^i \leftarrow_R \mathbb{Z}_p^{2\times 2}$ are sampled freshly. Since we are in the selective setting, $\mathcal{A}$ must choose all its encryption queries before making any functional key queries. For each encryption query of the form $(\boldsymbol{x}, \mathsf{id}, i)$, it receives the ciphertext $\mathsf{ct} = ([\![\boldsymbol{a}r]\!]_1, [\![(\boldsymbol{U}_0^i + \mathsf{id}\boldsymbol{U}_1^i)\boldsymbol{a}r]\!]_1, [\![\boldsymbol{V}^i\boldsymbol{a}r + \boldsymbol{x}]\!]_1)$ where the same randomness $r \leftarrow_R \mathbb{Z}_p$ is used to generate all ciphertexts. Recall that $\mathcal{A}$ can make at most one encryption query per instance $(\mathsf{pk}_i, \mathsf{msk}_i)$, which we denote by $(\boldsymbol{x}_i, \mathsf{id}_i^\star, i)$ — if it exists. Then, $\mathcal{A}$ sends queries of the form $([\![\boldsymbol{y}]\!]_2, \mathsf{id}, i)$ to $\mathcal{O}_{\mathsf{KeyGen}}$, upon which it gets $\mathsf{sk} = ([\![\boldsymbol{b}s]\!]_2, [\![\boldsymbol{V}^{i\top}\boldsymbol{y} + (\boldsymbol{U}_0^i + \mathsf{id}\boldsymbol{U}_1^i)^\top\boldsymbol{b}s]\!]_2, [\![\boldsymbol{y}]\!]_2)$, for fresh $s \leftarrow_R \mathbb{Z}_p$.

Game_1: we change the way ciphertexts are computed. Namely, each query $(\boldsymbol{x}_i, \mathsf{id}_i^\star, i)$ to the encryption oracle is now answered with

$$\mathsf{ct} = \left([\![\boldsymbol{z}]\!]_1, [\![(\boldsymbol{U}_0^i + \mathsf{id}_i^\star\boldsymbol{U}_1^i)\boldsymbol{z}]\!]_1, [\![\boldsymbol{V}^i\boldsymbol{z} + \boldsymbol{x}_i]\!]_1 \right) ,$$

where the same random coins $\boldsymbol{z} \leftarrow_R \mathbb{Z}_p^2$ are used to generate all ciphertexts. We prove that $\mathsf{Game}_0 \approx_c \mathsf{Game}_1$ by the DDH assumption in $\mathbb{G}_1$. Namely, we have $([\![\boldsymbol{a}]\!]_1, [\![\boldsymbol{a}r]\!]_1) \approx_c ([\![\boldsymbol{a}]\!]_1, [\![\boldsymbol{z}]\!]_1)$ where the leftmost distribution corresponds to Game_0, whereas the rightmost distribution corresponds to Game_1.

Game_2: we change the way ciphertexts are computed. Namely, each query $(\boldsymbol{x}_i, \mathsf{id}_i^\star, i)$ to the encryption oracle is now answered with

$$\mathsf{ct} = \left([\![\boldsymbol{z}]\!]_1, [\![(\boldsymbol{U}_0^i + \mathsf{id}_i^\star\boldsymbol{U}_1^i)\boldsymbol{z}]\!]_1, [\![\boldsymbol{V}^i\boldsymbol{z} + \boldsymbol{x}_i]\!]_1 \right) ,$$

where the same random coins $\boldsymbol{z} \leftarrow_R \mathbb{Z}_p^2 \setminus \mathsf{Span}(\boldsymbol{a})$ are used to generate all ciphertexts. Here $\mathsf{Span}(\boldsymbol{a})$ denotes the set of vectors proportional to $\boldsymbol{a}$. The cardinal of $\mathsf{Span}(\boldsymbol{a})$ is p, thus, the statistical distance between the uniform distribution over $\mathbb{Z}_p^2 \setminus \mathsf{Span}(\boldsymbol{a})$ and uniform over $\mathbb{Z}_p^2$ is $1/p$, and $\mathsf{Game}_1 \approx_s \mathsf{Game}_2$.

Game_3: we change the way the functional keys and challenge ciphertexts are computed. Namely, each query $(\boldsymbol{x}_i, \mathsf{id}_i^\star, i)$ to the encryption oracle is now answered with

$$\mathsf{ct} = \left([\![\boldsymbol{z}]\!]_1, [\![(\boldsymbol{U}_0^i + \mathsf{id}_i^\star\boldsymbol{U}_1^i)\boldsymbol{z}]\!]_1, [\![\boldsymbol{V}^i\boldsymbol{z}]\!]_1 \right) .$$

Note that this ciphertext does not depend on the message $\boldsymbol{x}_i$ anymore. Each query $([\![\boldsymbol{y}]\!]_2, \mathsf{id}, i)$ to $\mathcal{O}_{\mathsf{KeyGen}}$ is now answered with

$$\left([\![\boldsymbol{b}s]\!]_2, [\![\boldsymbol{V}^{i\top}\boldsymbol{y} - \boldsymbol{a}^\perp \cdot \boldsymbol{x}_i^\top\boldsymbol{y} + (\boldsymbol{U}_0^i + \mathsf{id}\boldsymbol{U}_1^i)^\top\boldsymbol{b}s]\!]_2, [\![\boldsymbol{y}]\!]_2 \right) ,$$

where $\boldsymbol{a}^\perp \in \mathbb{Z}_p^2$ is the vector such that $\boldsymbol{a}^\top\boldsymbol{a}^\perp = 0$ and $\boldsymbol{z}^\top\boldsymbol{a}^\perp = 1$. Game_1 and Game_2 are identically distributed, since for all i, all $\boldsymbol{x}_i \in \mathbb{Z}_p^d$, all $\boldsymbol{a}^\perp \in \mathbb{Z}_p^2$, the following are identically distributed: $\{\boldsymbol{V}^i \leftarrow_R \mathbb{Z}_p^{d\times 2} : \boldsymbol{V}^i\}$ and $\{\boldsymbol{V}^i \leftarrow_R \mathbb{Z}_p^{d\times 2} : \boldsymbol{V}^i - \boldsymbol{x}_i(\boldsymbol{a}^\perp)^\top\}$. The former distribution corresponds to Game_2, whereas the latter corresponds to Game_3. Note that here we are crucially relying on the fact that there is only one encryption

query per instance.

$\underline{\mathsf{Game}_4}$: we change the way the functional keys are computed. Namely, each query $([\![\boldsymbol{y}]\!]_2, \mathsf{id}, i)$ to $\mathcal{O}_{\mathsf{KeyGen}}$ is now answered with

$$\left([\![\boldsymbol{bs}]\!]_2, [\![\boldsymbol{V}^{i\top}\boldsymbol{y} - \mathbf{1}_{\mathsf{id}=\mathsf{id}_i^\star}\boldsymbol{a}^\perp\boldsymbol{x}_i^\top\boldsymbol{y} + (\boldsymbol{U}_0^i + \mathsf{id}\boldsymbol{U}_1^i)^\top\boldsymbol{bs}]\!]_2, [\![\boldsymbol{y}]\!]_2\right) .$$

That is, now we only add the terms $\boldsymbol{a}^\perp\boldsymbol{x}_i^\top\boldsymbol{y}$ for functional key queries $(\boldsymbol{y}, \mathsf{id}, i)$ where $\mathsf{id} = \mathsf{id}_i^\star$, i.e. the identity matches that of the ciphertext for instance $(\mathsf{pk}_i, \mathsf{msk}_i)$ — if there is no ciphertext queried for instance i then $\mathbf{1}_{\mathsf{id}=\mathsf{id}_i^\star}$ is set to 0. To transition from Game_3 to Game_4, we use the following hybrid games.

$\underline{\mathsf{Game}_{3.j}}$: for all $j \in \{0, \ldots, Q\}$, where Q denotes the number of functional key queries, $\mathsf{Game}_{3.j}$ is defined as Game_4 for the first j'th key queries and as Game_3 for the last $Q - j$ queries. By definition we have $\mathsf{Game}_3 = \mathsf{Game}_{3.0}$ and $\mathsf{Game}_4 = \mathsf{Game}_{3.Q}$. It suffices to show that for all $j \in [Q]$, $\mathsf{Game}_{3.j-1} \approx_c \mathsf{Game}_{3.j}$. To do so, we introduce new intermediate games, defined as follows.

$\underline{\mathsf{Game}_{3.j-1.1}}$: is defined as $\mathsf{Game}_{3.j-1}$, except the j'th query to $\mathcal{O}_{\mathsf{KeyGen}}$, denoted by $([\![\boldsymbol{y}_j]\!]_2, \mathsf{id}_j, i_j)$, is now answered with

$$\left([\![\boldsymbol{d}]\!]_2, [\![\boldsymbol{V}^{i_j\top}\boldsymbol{y}_j - \boldsymbol{a}^\perp\boldsymbol{x}_{i_j}^\top\boldsymbol{y}_j + (\boldsymbol{U}_0^{i_j} + \mathsf{id}_j\boldsymbol{U}_1^{i_j})^\top\boldsymbol{d}]\!]_2, [\![\boldsymbol{y}_j]\!]_2\right) ,$$

where $\boldsymbol{d} \leftarrow_R \mathbb{Z}_p^2$. We have $\mathsf{Game}_{3.j-1} \approx_c \mathsf{Game}_{3.j-1.1}$ by the DDH assumption in $\mathbb{G}_2$, which states that $([\![\boldsymbol{b}]\!]_2, [\![\boldsymbol{bs}_j]\!]_2) \approx_c ([\![\boldsymbol{b}]\!]_2, [\![\boldsymbol{d}]\!]_2)$ where $\boldsymbol{b}, \boldsymbol{d} \leftarrow_R \mathbb{Z}_p^2, s_j \leftarrow_R \mathbb{Z}_p$. The former distribution corresponds to $\mathsf{Game}_{3.j-1}$, whereas the latter corresponds to $\mathsf{Game}_{3.j-1.1}$.

$\underline{\mathsf{Game}_{3.j-1.2}}$: is defined as $\mathsf{Game}_{3.j-1.1}$, except the vector $\boldsymbol{d}$ used to compute the j'th queried functional secret key is sampled as $\boldsymbol{d} \leftarrow_R \mathbb{Z}_p^2 \setminus \mathsf{Span}(\boldsymbol{b})$, instead of uniformly random over $\mathbb{Z}_p^2$. Since the cardinal of $\mathsf{Span}(\boldsymbol{b})$ is at most p, the uniform distribution over $\mathbb{Z}_p^2 \setminus \mathsf{Span}(\boldsymbol{b})$ has statistical distance at most $1/p$ with the uniform distribution over $\mathbb{Z}_p^2$. Thus, $\mathsf{Game}_{3.j-1.1} \approx_s \mathsf{Game}_{3.j-1.2}$.

$\underline{\mathsf{Game}_{3.j-1.3}}$: is defined as $\mathsf{Game}_{3.j-1.2}$, except the j'th query to $\mathcal{O}_{\mathsf{KeyGen}}$ is now answered with

$$\left([\![\boldsymbol{d}]\!]_2, \left[\!\!\left[\boldsymbol{V}^{i_j\top}\boldsymbol{y}_j - \mathbf{1}_{\mathsf{id}_j=\mathsf{id}_{i_j}^\star}\boldsymbol{a}^\perp\boldsymbol{x}_{i_j}^\top\boldsymbol{y}_j + (\boldsymbol{U}_0^{i_j} + \mathsf{id}_j\boldsymbol{U}_1^{i_j})^\top\boldsymbol{d}\right]\!\!\right]_2, [\![\boldsymbol{y}_j]\!]_2\right) ,$$

where $\boldsymbol{d} \leftarrow_R \mathbb{Z}_p^2 \setminus \mathsf{Span}(\boldsymbol{b})$. Note that if $\mathsf{id}_{i_j}^\star = \mathsf{id}_j$, then the two games $\mathsf{Game}_{3.j-1.2}$ and $\mathsf{Game}_{3.j-1.3}$ are identical. Thus we focus on the case $\mathsf{id}_{i_j}^\star \neq \mathsf{id}_j$. In that case we show that $\mathsf{Game}_{3.j-1.3}$ is also identically distributed to $\mathsf{Game}_{3.j-1.2}$ using a statistical argument, which roughly says that the vectors $\boldsymbol{U}_0^{i_j}\boldsymbol{b}$ and $\boldsymbol{U}_0^{i_j}\boldsymbol{d}$ are statistically independent since $\boldsymbol{b}$ and $\boldsymbol{d}$ are linearly independent. The same holds with respect to the matrix $\boldsymbol{U}_1^{i_j}$. Thus, because the vectors $\boldsymbol{U}_0^{i_j}\boldsymbol{d}$ and $\boldsymbol{U}_1^{i_j}\boldsymbol{d}$ are fresh and used only for the j'th functional secret key, we can use a pairwise independence argument to conclude. More formally, we use the fact that for all $\mathsf{id}_j, \mathsf{id}_{i_j}^\star \in \mathbb{Z}_p$ such that $\mathsf{id}_{i_j}^\star \neq \mathsf{id}_j$, all vectors $\boldsymbol{a}^\perp, \boldsymbol{b}^\perp \in \mathbb{Z}_p^2$, the following distributions are the same:

$$\left\{\boldsymbol{U}_0^{i_j}, \boldsymbol{U}_1^{i_j} \leftarrow_R \mathbb{Z}_p^{2\times 2} : (\boldsymbol{U}_0^{i_j}, \boldsymbol{U}_1^{i_j})\right\}$$

$$\text{and}\left\{\boldsymbol{U}_0^{i_j}, \boldsymbol{U}_1^{i_j} \leftarrow_R \mathbb{Z}_p^{2\times 2} : \left(\boldsymbol{U}_0^{i_j} + \frac{-\mathsf{id}_{i_j}^\star\boldsymbol{x}_{i_j}^\top\boldsymbol{y}_j\boldsymbol{b}^\perp(\boldsymbol{a}^\perp)^\top}{\mathsf{id}_j - \mathsf{id}_{i_j}^\star}, \boldsymbol{U}_1^{i_j} + \frac{\boldsymbol{x}_{i_j}^\top\boldsymbol{y}_j\boldsymbol{b}^\perp(\boldsymbol{a}^\perp)^\top}{\mathsf{id}_j - \mathsf{id}_{i_j}^\star}\right)\right\} .$$

The former distribution corresponds to $\mathsf{Game}_{3.j-1.2}$ (with pre and post-processing), whereas the latter distribution corresponds to $\mathsf{Game}_{3.j-1.3}$ (with the same pre and post processing).

$\underline{\mathsf{Game}_{3.j-1.4}}$: is defined as $\mathsf{Game}_{3.j-1.3}$, except the vector $\boldsymbol{d}$ used to compute the j'th queried functional secret key is sampled $\boldsymbol{d} \leftarrow_R \mathbb{Z}_p^2$, instead of uniformly random over $\mathbb{Z}_p^2 \setminus \mathsf{Span}(\boldsymbol{b})$. This is the reverse to the transition from $\mathsf{Game}_{3.j-1.2}$ to $\mathsf{Game}_{3.j-1.3}$. By the same statistical argument, we obtain $\mathsf{Game}_{3.j-1.3} \approx_s \mathsf{Game}_{3.j-1.4}$.

Finally, note that $\mathsf{Game}_{3.j-1.4}$ is the same as $\mathsf{Game}_{3.j}$ except the j'th queried key is computed using $[\![\boldsymbol{d}]\!]_2 \leftarrow_R \mathbb{G}_2^2$ in the former, and $[\![\boldsymbol{b}s_j]\!]_2 \in \mathbb{G}_2^2$ with $s_j \leftarrow_R \mathbb{Z}_p$ in the latter. Therefore, we have $\mathsf{Game}_{3.j-1.4} \approx_c \mathsf{Game}_{3.j}$ by the DDH assumption, which states that $([\![\boldsymbol{b}]\!]_2, [\![\boldsymbol{d}]\!]_2) \approx_c ([\![\boldsymbol{b}]\!]_2, [\![\boldsymbol{b}s_j]\!]_2)$ where $\boldsymbol{b}, \boldsymbol{d} \leftarrow_R \mathbb{Z}_p^2, s_j \leftarrow_R \mathbb{Z}_p$. The former distribution corresponds to $\mathsf{Game}_{3.j-1.4}$, whereas the latter distribution corresponds to $\mathsf{Game}_{3.j}$. Note that this transition is exactly reverse to the transition from $\mathsf{Game}_{3.j-1.1}$ to $\mathsf{Game}_{3.j-1.2}$. This concludes the proof that $\mathsf{Game}_{3.j-1} \approx_c \mathsf{Game}_{3.j}$ and consequently, that $\mathsf{Game}_3 \approx_c \mathsf{Game}_4$.

Note that Game_4 exactly corresponds to the ideal experiment from the security definition in Section 3.1 for the simulator defined in Fig. 6, which concludes the proof. □

$\overline{\mathsf{GlobalSetup}}(1^\lambda)$:
$(\boldsymbol{a}|\boldsymbol{z}) \leftarrow_R \mathsf{GL}_2$, let $\boldsymbol{a}^\perp \in \mathbb{Z}_p^2$ be a vector such that $\boldsymbol{a}^\top \boldsymbol{a}^\perp = 0 \wedge \boldsymbol{z}^\top \boldsymbol{a}^\perp = 1$; set $\boldsymbol{b} \leftarrow_R \mathbb{Z}_p^2$.
For every $i \in [n]$, sample $\boldsymbol{U}_0^i, \boldsymbol{U}_1^i \leftarrow_R \mathbb{Z}_p^{2\times 2}$, $\boldsymbol{V}^i \leftarrow_R \mathbb{Z}_p^{d\times 2}$.
Set $\mathsf{st} := (\boldsymbol{V}^i, \boldsymbol{U}_0^i, \boldsymbol{U}_1^i)_{i\in[n]}$.
Return $(\mathsf{st}, \overline{\mathsf{gp}})$.

$\overline{\mathsf{Enc}}\left(\mathsf{st}, \overline{\mathsf{pk}_i}, \mathsf{id}_i^\star, i, \mathit{leakage}\right)$:
(Note that $\mathit{leakage} = \emptyset$.[a]) Return $\mathsf{ct} := ([\![\boldsymbol{z}]\!]_1, [\![(\boldsymbol{U}_0^i + \mathsf{id}_i^\star \boldsymbol{U}_1)\boldsymbol{z}]\!]_1, [\![\boldsymbol{V}^i \boldsymbol{z}]\!]_1)$.

$\overline{\mathsf{KeyGen}}\left(\mathsf{st}, [\![\boldsymbol{y}_j]\!]_2, \mathsf{id}_j, i_j, \mathit{leakage}\right)$:
(Note that $\mathit{leakage}$ contains the value $\left[\!\left[\boldsymbol{x}_{i_j}^\top \boldsymbol{y}_j\right]\!\right]_2$ if $\mathsf{id}_j = \mathsf{id}_{i_j}^\star$.)
Sample $s_j \leftarrow_R \mathbb{Z}_p$ and return:

$$\left([\![\boldsymbol{b}s_j]\!]_2, [\![\boldsymbol{V}^{i_j\top} \boldsymbol{y}_j + \mathbf{1}_{\mathsf{id}_j \neq \mathsf{id}_{i_j}^\star} \boldsymbol{a}^\perp \cdot \boldsymbol{x}_{i_j}^\top \boldsymbol{y}_j + (\boldsymbol{U}_0^{i_j} + \mathsf{id}_j \boldsymbol{U}_1^{i_j})^\top \boldsymbol{b}s_j]\!]_2\right) \ .$$

[a] Since we are in the selective setting, no queries to $\mathcal{O}_{\mathsf{KeyGen}}$ can be made before all queries to $\mathcal{O}_{\mathsf{Enc}}$ have been sent.

Fig. 6. Simulator for the security proof of the ID-IPFE from Fig. 2.