Generic Negation of Pair Encodings

Miguel Ambrona

NTT Secure Platform Laboratories, Tokyo, Japan
miguel.ambrona.fu@hco.ntt.co.jp

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. Pair encodings (Attrapadung, EUROCRYPT 2014) are simple primitives that can be used for constructing fully secure ABE schemes associated to a predicate relative to the encoding. We propose a generic transformation that takes any pair encoding scheme (PES) for a predicate P and produces a PES for its negated predicate $\overline{P}$. This construction finally solves a problem that was open since 2015. Our techniques bring new insight to the expressivity and generality of PES and can be of independent interest. We also provide, to the best of our knowledge, the first pair encoding scheme for negated doubly spatial encryption (obtained with our transformation) and explore several other consequences of our results.

1 Introduction

Attribute-based encryption (ABE) is a form of public-key encryption that generalizes the traditional single-recipient variant, providing fine-grained access control on the encrypted data. In this new paradigm, ciphertexts and keys have attributes attached and the decryption ability of a key on a ciphertext is determined by a potentially complex access control policy involving these attributes. More concretely, an ABE scheme for predicate P guarantees that the decryption of a ciphertext ct_x with a secret key sk_y is successful if and only if the ciphertext attribute x and the key attribute y verify the predicate, i.e., $P(x, y) = 1$.

ABE was first conceived by Sahai and Waters [SW05] and later introduced by Goyal *et al.* [GPSW06]. Originally, ABE was designed in the flavour of key-policy ABE (KP-ABE), where value x is a Boolean vector, value y is a Boolean function and predicate $P(x, y)$ is defined as $y(x) \stackrel{?}{=} 1$. On the other hand, in the analogous version, ciphertext-policy ABE (CP-ABE), the roles of values x and y are swapped. Nowadays, the notion of ABE has been generalized and, thanks to a considerable effort by the community of cryptographers, there exist efficient schemes for a rich variety of predicates. For example, identity-based encryption (IBE) [Sha84] can be obtained as $P(x, y) \coloneqq x \stackrel{?}{=} y$, zero-inner product encryption (ZIPE) [KSW08] can be obtained by setting $P(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \langle \boldsymbol{x}, \boldsymbol{y} \rangle \stackrel{?}{=} 0$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ belong to some vector space; other examples are span programs [KW93], non-monotonic access structures [OSW07], hierarchical IBE [LW11], large universe ABE [RW13], polynomial size circuits [GVW13], or regular languages [Wat12].

Despite such a great progress in the field, designing better schemes in terms of size, performance, security and expressivity became an excessively hard and tedious task. Until two astonishing works appeared in 2014.

Modular frameworks for ABE. In 2014, Wee [Wee14] and Attrapadung [Att14] independently proposed two generic and unifying frameworks for designing attribute-based encryption schemes for different predicates. Both works define a simple primitive called *encoding* and follow the dual system methodology by Lewko and Waters [LW10, Wat09] to construct a compiler that, on input an encoding (for certain predicate P), produces a fully secure attribute-based encryption scheme for P. Wee defines so-called *predicate encodings*, an information-theoretic primitive inspired by linear secret sharing, while Attrapadung introduces the notion of *pair encodings*, a similar primitive that admits both information-theoretic and computational security definitions. These frameworks remarkably simplify the design and study of ABE schemes: the designer can focus on the construction of the simpler encoding (for the desired predicate), which requires weaker security properties that are more easily verifiable. In fact, the potential of this new frameworks is evidenced by the invention of new constructions and performance improvements on existing primitives. Although these frameworks were designed over composite-order groups, they were both extended, in [CGW15] and [Att16] respectively, to the prime-order setting (under the Matrix-DH assumption). Subsequent works propose variations and extensions of these modular frameworks [AC16, AC17, CMP17], some of them even redefining the core encoding primitive [KSGA16] (defining so-called *tag-based* encodings). However, note that the frameworks based on *pair encodings* are the most general and expressive[1] and they have led to breakthrough constructions such as constant-size ciphertext KP-ABE (with large universes) [Att14], fully-secure functional encryption for regular languages [Att14], constant-size ciphertext CP-ABE [AC16] or completely-unbounded KP-ABE for non-monotone span programs (NSP) over large universes [Att19]. Note that, even nowadays, it is still unknown how to construct any of these powerful schemes based on predicate encodings or tag-based encodings.

Generic predicate transformations. In order to further simplify the design of these encodings, a common practice is to develop techniques to modify or combine existing ones. For example, the *DUAL* transformation, that swaps the ciphertext attribute and the key attribute, or the *AND* transformation, that joins two predicates in conjunction, can be achieved for pair encodings [Att14, AY15]. Among many applications, these transformations can be used to build dual-policy attribute-based encryption (DP-ABE) [AI09, AY15]; or to enhance any encoding with direct revocation of keys by combining (in conjunction) the original encoding with, e.g., an encoding for broadcast encryption.

In the framework of [CGW15], Ambrona *et al.* [ABS17] designed new general transformations for the *DUAL*, *OR* and *AND* connectors and, remarkably, the

[1] In fact, it is known that *predicate encodings* are a subclass of *pair encodings* [ABS17].

NOT transformation (that negates the predicate of the encoding). This functionally complete set of Boolean transformers provides a rich combination of predicates and arguably broadens the expressivity of the framework, however, such a negation is *limited to the framework based on predicate encodings*. Designing a similar negation transformation that is applicable to all pair encoding schemes (PES) is a very appealing problem, since it would facilitate the design of new encodings and would immediately expand the expressivity of the PES framework by applying it to all existing ones. Note that, as we have already mentioned, pair encodings have proven themselves to be significantly more expressive than any other related framework.

However, recent works have considered the problem of designing such a general negation to be intrinsically hard [AC17, Att19] (see our discussion in Section 3, we also refer to this section for more details about relevant related works). To the best of our knowledge, a general *NOT* transformation that is applicable to the framework of pair encodings does not exist in the literature.

1.1 Our contribution

We pursue the study of pair encoding schemes and establish several general results that can lead to performance improvements, and new encodings that broaden their scope.

Generic negation of pair encodings. We propose a generic transformation that takes any pair encoding scheme for a predicate P and produces a pair encoding scheme for its negated predicate, $\overline{P}$. Our transformation is applicable to pair encodings that follow the most recent and refined definition given in [AC17]. Our construction finally solves a problem that was open since 2015, when several other transformation for pair encodings (like conjunction or duality) were proposed [AY15], but no generic negation was provided (nor designed in subsequent works). In fact, several works had suggested that finding such a transformation was non-obvious [AC17, Att19], since it relates to the problem of generically finding a short "certificate" of security of the encoding. We elaborate on this idea in Section 3.

Algebraic characterization of pair encodings. En route to designing our generic negation, we define an algebraic characterization of PES that brings new insight to their expressivity and generality and can be of independent interest. Our characterization allows us to express the security of a pair encoding scheme as the (in)existence of solutions to a system of matrix equations. This is the bridge that allows us to leverage Lemma 1, a very powerful result from linear algebra (commonly used in cryptography), in order to design and prove our generic negation.

New encodings. Our generic negation facilitates the design of new pair encoding schemes. It will immediately provide us with a negated version of any encoding, something particularly useful for encodings for which a negated counterpart is

not known. A relevant example of a PES with (previously) unknown negation is the case of doubly spatial encryption.

Doubly spatial encryption [Ham11] is an important primitive that generalizes both spatial encryption and *negated spatial encryption* [AL10]. A negated doubly spatial encryption scheme serves as its revocation analogue and can lead to powerful generalizations in the same way that negated (standard) spatial encryption unifies existing primitives, e.g. it subsumes *non-zero-mode* inner-product encryption (IPE) [AL10]. In Section 6.1 we provide, to the best of our knowledge, the first pair encoding scheme for negated doubly spatial encryption, obtained with our transformation.

Other implications of our results. We believe the results presented in this work improve our understanding of pair encodings and how expressive they are. In particular, we now know that *the set of predicates that can be expressed with PES is closed under negation.* In Section 6.2, we elaborate on the conclusions we could derive from this fact as well as discuss how our generic transformation can also lead to performance improvements when implementing ABE schemes. Furthermore, note that our generic negation is compatible with the very recent framework proposed by Attrapadung [Att19], designed to perform dynamic pair encoding compositions. We believe our new transformation complements his work, where the proposed non-monotone formulae composition was only *semi-generic* (but dynamic), because he had to rely on encodings for which a negated version was available.

2 Preliminaries

2.1 Notation

We write $s \xleftarrow{\$} S$ to denote that s is uniformly sampled from a set S. For integers m, n, we define $[m, n]$ as the range $\{m, \ldots, n\}$ and we denote by $[n]$ the range $[1, n]$. We use the same conventions for matrix-representations of linear maps on finite-dimensional spaces. For a ring R, we define vectors $\boldsymbol{v} \in R^n$ as *column* matrices, denote the transpose of a matrix A by $A^\top$ and its trace by $\mathrm{tr}(A)$. We denote by $|\boldsymbol{v}|$ the length or dimension of vector $\boldsymbol{v}$ and by v_i its i-th component, for all $i \in \{1, \ldots, |\boldsymbol{v}|\}$. Similarly, A_i denotes the i-th *row* of matrix A (we do not use this notation when the name of the matrix already contains a subindex). We denote by $\mathrm{span}(A)$ the linear *column span* of matrix A. We denote the identity matrix of dimension n by I_n, a zero vector of length n by $\mathbf{0}_n$ and a zero matrix of m rows and n columns by $0_{m\times n}$. We denote by $\boldsymbol{e}_i^n$ the i-th vector of the standard basis of an n-dimensional space, for all $i \in [n]$. We sometimes denote $\boldsymbol{e}_1^n$ by $\mathbf{1}_n$. Similarly, we denote by $1_{m\times n}$ the matrix $\mathbf{1}_m \mathbf{1}_n^\top$, i.e., *a null matrix of m rows and n columns whose component in the first row and first column is* 1. Given two matrices A and B, we denote by $A \otimes B$ their Kronecker product.

We consider a bilinear group generator $\mathcal{G}$ that takes a security parameter $\lambda \in \mathbb{N}$ and outputs the description of a bilinear group $(p, G_1, G_2, G_\mathsf{t}, g_1, g_2, e)$ where G_1, G_2 and G_t are cyclic groups of order p (for a λ-bits prime p), g_1

and g_2 are generators of G_1 and G_2 respectively and $e : G_1 \times G_2 \to G_\mathsf{t}$ is a (non-degenerate) bilinear map, satisfying $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $a, b \in \mathbb{N}$. Observe that the element $g_\mathsf{t} = e(g_1, g_2)$ generates G_t.

2.2 Attribute-based encryption

Attribute-based encryption (ABE) [SW05] is a form of of public-key encryption that supports fine-grained access control of encrypted data.

Definition 1 (Attribute-based encryption). *An ABE scheme for predicate $P : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ consists of four probabilistic polynomial-time algorithms:*

- $\mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}) \to (\mathsf{mpk}, \mathsf{msk})$*, on input the security parameter λ and attribute universes $\mathcal{X}, \mathcal{Y}$, outputs a master public key and a master secret key, defining a key space $\mathcal{K}$.*
- $\mathsf{Enc}(\mathsf{mpk}, x) \to (\mathsf{ct}_x, \tau)$*, on input* mpk *and a ciphertext attribute $x \in \mathcal{X}$, outputs a ciphertext ct_x and a symmetric encryption key $\tau \in \mathcal{K}$.*
- $\mathsf{KeyGen}(\mathsf{msk}, y) \to \mathsf{sk}_y$*, on input the master secret key and a key attribute $y \in \mathcal{Y}$, outputs a secret key sk_y.*
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{ct}_x, x) \to \tau/\perp$*, on input sk_y and ct_x, outputs a symmetric key $\tau \in \mathcal{K}$ if $P(x, y) = 1$ or $\perp$ otherwise.*

Correctness. For all $\lambda \in \mathbb{N}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $P(x, y) = 1$, it holds:

$$\Pr\left[\begin{array}{r}(\mathsf{msk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{msk}, y) \\ (\mathsf{ct}_x, \tau) \leftarrow \mathsf{Enc}(\mathsf{mpk}, x)\end{array} : \mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{ct}_x, x) = \tau\right] = 1 \ .$$

Security. Informally, an ABE scheme is secure if no probabilistic polynomial-time (PPT) adversary can distinguish the symmetric encryption key associated to a ciphertext $\mathsf{ct}_{x^\star}$ (for some attribute $x^\star$) from a uniformly chosen one from $\mathcal{K}$, even after requesting several secret keys for attributes y of their choice, as long as they all satisfy $P(x^\star, y) = 0$.

In this work we focus on *pair encodings* (see the next section) as a building block for constructing ABE schemes and we refer to Appendix B.1 for a formal security definition of ABE, which we do not state here. Instead, we will formally state and reason about the security requirements for pair encodings.

2.3 Pair encodings

We consider the refined definition of pair encodings introduced by Agrawal and Chase in [AC17].

Definition 2 (Pair encoding). *A* pair encoding scheme *(PES) for a predicate family $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ indexed by $\kappa = (N, \mathsf{par})$ consists of the following deterministic and efficiently computable algorithms:*

- $\mathsf{Param}(\mathsf{par})$*: on input certain parameters outputs an integer n, specifying the number of* common variables*, denoted by $\boldsymbol{b} = (b_1, \ldots, b_n)$.*
- $\mathsf{EncKey}(N, y)$*: on input $N \in \mathbb{N}$ and $y \in \mathcal{Y}_{(N,\mathsf{par})}$, outputs a vector of polynomials $\boldsymbol{k} = (k_1, \ldots, k_{m_3})$ in the* non-lone *variables $\boldsymbol{r} = (r_1, \ldots, r_{m_1})$, the* lone *variables $\hat{\boldsymbol{r}} = (\alpha, \hat{r}_1, \ldots, \hat{r}_{m_2})$ and the common variables $\boldsymbol{b}$.*
- $\mathsf{EncCt}(N, x)$*: on input $N \in \mathbb{N}$ and $x \in \mathcal{X}_{(N,\mathsf{par})}$, outputs a vector of polynomials $\boldsymbol{c} = (c_1, \ldots, c_{w_3})$ in the* non-lone *variables $\boldsymbol{s} = (s_0, s_1, \ldots, s_{w_1-1})$, the* lone *variables $\hat{\boldsymbol{s}} = (\hat{s}_1, \ldots, \hat{s}_{w_2})$ and the common variables $\boldsymbol{b}$.*
- $\mathsf{Pair}(N, x, y)$*: on input $N \in \mathbb{N}$ and attributes x and y, outputs a pair of matrices (E, E') with coefficients in $\mathbb{Z}_N$ of dimensions $w_1 \times m_3$ and $w_3 \times m_1$ respectively.*

We require that the following properties be satisfied:

reconstructability: *For every $\kappa = (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $P_\kappa(x, y) = 1$, the following equation holds symbolically:*

$$\boldsymbol{s}^\top E \boldsymbol{k} + \boldsymbol{c}^\top E' \boldsymbol{r} = \alpha s_0 \ ,$$

where $\boldsymbol{k} \leftarrow \mathsf{EncKey}(N, x)$, $\boldsymbol{c} \leftarrow \mathsf{EncCt}(N, y)$ and $(E, E') \leftarrow \mathsf{Pair}(N, x, y)$.

structural constraints: *The polynomials produced by* EncKey *only contain monomials of the form α, $r_i b_j$ or $\hat{r}_{i'}$ for some $i \in [m_1]$, $j \in [n]$ and $i' \in [m_2]$. On the other hand, the polynomials produced by* EncCt *only contain monomials of the form $s_i b_j$ or $\hat{s}_{i'}$ for some $i \in [0, w_1 - 1]$, $j \in [n]$ and $i' \in [w_2]$.*

security (non-reconstructability): *For all $\kappa \in (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $P_\kappa(x, y) = 0$, and for every pair of matrices E and E', over $\mathbb{Z}_N$, $\boldsymbol{s}^\top E \boldsymbol{k} + \boldsymbol{c}^\top E' \boldsymbol{r} \neq \alpha s_0$, where $\boldsymbol{k} \leftarrow \mathsf{EncKey}(N, x)$ and $\boldsymbol{c} \leftarrow \mathsf{EncCt}(N, y)$.*

Remark 1. Observe that m_1 and w_1 represent[2] the number of non-lone variables $\boldsymbol{r}$ and $\boldsymbol{s}$ respectively; m_2 and w_2 represent the number of lone variables $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{s}}$ respectively; and m_3 and w_3 represent the number of polynomials produced by EncKey and EncCt respectively. Also note that m_3 may depend on the key attribute y and w_3 may depend on the ciphertext attribute x. We will use this notation throughout the paper.

Agrawal and Chase [AC17] showed that an encoding with the non-reconstructability property (coined *non-trivially broken*) satisfies the *symbolic property*, a

[2] In some literature, the number of non-lone ciphertext variables is defined as w_1+1, since the special variable s_0 is treated separately. Observe that our vector of non-lone variables ranges from s_0 to s_{w_1-1}, this is for the sake of notation in further sections.

concept introduced by them which is a sufficient condition to build attribute-based encryption in the standard model under the so-called q-ratio assumption.

We refer to Appendix B.2 for details about how the compiler from PES to fully secure ABE works. In this work we directly reason about PES and do not need to explicitly define such a compiler. However, for the sake of understanding, we provide an intuition of how a PES can be used to create an ABE scheme in the following section.

Example 1 (PES for identity-based encryption). The following is a pair encoding scheme for the IBE predicate $P(x, y) := x \stackrel{?}{=} y$, for $x, y \in \mathbb{Z}_N$. (With $m_1 = 1$, $m_2 = 0$, $m_3 = 2$ and $w_1 = 2$, $w_2 = 0$, $w_3 = 1$.)

$$\mathsf{EncKey}(N, y) := \{\alpha + r_1 b_1,\ r_1(yb_2 + b_3)\} \quad \mathsf{EncCt}(N, x) := \{s_0 b_1 + s_1(xb_2 + b_3)\}\ .$$

Furthermore, in this case Param is an algorithm that simply outputs $n = 3$ and $\mathsf{Pair}(N, x, y)$ returns matrices $E = I_2$ and $E' = -I_1$. For reconstructability, observe that $\begin{pmatrix} s_0 & s_1 \end{pmatrix} E\boldsymbol{k} + \boldsymbol{c}^\top E' \begin{pmatrix} r_1 \end{pmatrix}$ equals $s_0\alpha + s_1 r_1(yb_2 + b_3) - s_1 r_1(xb_2 + b_3)$ which equals αs_0 whenever $x = y$, as desired.

Arguing security, i.e., non-reconstructability whenever $x \neq y$, is a little trickier. One needs to show that for all matrices $E \in \mathbb{Z}_N^{2\times 2}$, $E' \in \mathbb{Z}_N$, the above linear combination is never equal to αs_0. This could be done by unfolding the list of polynomials in $\boldsymbol{s} \otimes \boldsymbol{k}$, $\boldsymbol{c} \otimes \boldsymbol{r}$ into a matrix A with $w_1 m_3 + w_3 m_1$ rows (as many as polynomials) and as many columns as different monomials appear in them, where the element at row i and column j of the matrix represents the coefficient of the j-th monomial in the i-th polynomial. (Let the first column be the one associated to monomial αs_0.) One could then argue security by checking that the row span of A does not contain the vector $(1\,0\,\ldots\,0)$ when $P(x, y) = 0$.

However, there is a simpler way of proving non-reconstructability. Simply evaluate the polynomials produced by EncKey and EncCt in:

$$b_1 \leftarrow -1 \qquad b_2, s_0, r_1, \alpha \leftarrow 1 \qquad b_3 \leftarrow -y \qquad s_1 \leftarrow (x-y)^{-1}\ .$$

Since all the polynomials evaluate to 0, but αs_0 evaluates to $1 \neq 0$, it must be impossible to symbolically reconstruct αs_0 with some pair of matrices E, E'. Otherwise, we would have a contradiction:

$$0 \;=\; \boldsymbol{s}^\top E \mathbf{0}_{m_3} + \mathbf{0}_{w_3}^\top E' \boldsymbol{r} \;=\; \boldsymbol{s}^\top E \boldsymbol{k}(\boldsymbol{r}, \boldsymbol{b}) + \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{b})^\top E' \boldsymbol{r} \;=\; \alpha s_0 \;=\; 1\ .$$

The above variable substitution that vanishes all polynomials, but does not vanish polynomial αs_0 can be considered to be a short "certificate" of the security of the scheme (and it is well-defined as long as $x \neq y$). We elaborate on this interesting method for arguing security in Section 3. ■

2.4 ABE from PES

The compiler from pair encodings to attribute-based encryption is defined over bilinear groups implemented as dual system groups (DSG) [CW13, CW14, AC17].

Here, we define a simplified version of the compiler and avoid DSG for simplicity, but note that the actual scheme produced by these compilers uses vectors of group elements where we write single group elements. We provide a complete description of the compiler from [AC17] in Appendix B.2.

Informally, the symmetric encryption key is computed as $\tau := g_t^{\alpha s_0}$, where s_0 is fresh randomness and g_t^{α} is part of the master public key. Both ciphertexts and keys are made of group elements (created based on the recipe given by the corresponding PES polynomials). It is possible to recover τ when the predicate is satisfied. More concretely, for $\boldsymbol{k} \leftarrow \mathsf{EncKey}(x)$ and $\boldsymbol{c} \leftarrow \mathsf{EncCt}(y)$, the compiler could be summarized as follows:

$$\mathsf{mpk} := \{g_t^{\alpha},\, g_1^{\boldsymbol{b}}\} \qquad (\mathsf{ct}_x, \tau) := \left(\{g_1^{\boldsymbol{s}},\, g_1^{\boldsymbol{c}(\boldsymbol{s},\hat{\boldsymbol{s}},\boldsymbol{b})}\}, g_t^{\alpha s_0}\right)$$
$$\mathsf{msk} := \{\alpha,\, \boldsymbol{b}\} \qquad \mathsf{sk}_y := \{g_2^{\boldsymbol{r}},\, g_2^{\boldsymbol{k}(\boldsymbol{r},\hat{\boldsymbol{r}},\boldsymbol{b})}\}$$

Decryption is done by pairing $g_1^{\boldsymbol{s}}$ with $g_2^{\boldsymbol{k}}$, $g_1^{\boldsymbol{c}}$ with $g_2^{\boldsymbol{r}}$, and linearly combining the resulting elements, according to the coefficients given by $\mathsf{Pair}(x, y)$, obtaining αs_0 in the exponent.

2.5 Linear algebra tools

In order to prove the validity of our generic negation of pair encodings, we will use a very powerful result from linear algebra that has been widely used in the literature [Bei11, AC16, AC17, ABS17]. It states that given a field K, a matrix $A \in K^{m\times n}$ and a vector $\boldsymbol{z} \in K^m$, it holds that $A\boldsymbol{v} \neq \boldsymbol{z}$ for all $\boldsymbol{v} \in K^n$ if and only if there exists a vector $\boldsymbol{w} \in K^m$ such that $\boldsymbol{w}^\top A = \boldsymbol{0}_n$ and $\boldsymbol{w}^\top \boldsymbol{z} = 1$. We refer to [Bei11, Claim 2] for a formal proof.

Here, for the sake of presentation, we state a variant of the above result, which can be shown to be equivalent, but that facilitates its application in the proof of Lemma 2.

Lemma 1. *Let V and W be vector spaces over a field K. Let $f : V \to W$ be a linear operator and let $\boldsymbol{z} \in W$. We have that:*

$$\boldsymbol{z} \notin \mathrm{Im}(f) \quad \Leftrightarrow \quad \exists \varphi \in W^* \text{ such that } \varphi \circ f = 0 \ \wedge\ \varphi(\boldsymbol{z}) = 1 \ .$$

Here, W^* denotes the dual space *of W, i.e., the set of all linear maps $\varphi : W \to K$.*

3 Overview of our generic negation transformation

Our starting point is the generic negation for the (less expressive) framework of *predicate encodings* from [ABS17]. In order to achieve their transformation, Ambrona, Barthe, and Schmidt first defined an algebraic characterization of predicate encodings where the security of the encoding (previously defined as an equality between distributions) was redefined into a purely algebraic statement related to the existence of solutions to a linear system of equations. This observation allowed them to link the notions of security and non-reconstructability and

define what they coined the *implicit predicate* of an encoding. This implies, in a nutshell, that all functions mapping attributes into matrices define a valid predicate encoding for a certain predicate, informally defined as all pairs of attributes (x, y) that map into matrices that lead to reconstructability.

Now that security has been proven to be equivalent to non-reconstructability, and given the simple structure of predicate encodings (which are essentially matrices over $\mathbb{Z}_p$), it is possible to find a short "witness" of non-reconstructability by simply finding a solution to a dual system of equations.[3] What we want to highlight here is that their new understanding of predicate encodings allows them to view both reconstructability and non-reconstructability as essentially the same kind of property. This suggests that one may be able to build a generic negation of predicate encodings by transposing the matrices induced by them.[4] This is in fact what the negation by Ambrona *et al.* does, but extra care is needed to make things really work.

Unfortunately, in the case of pair encodings things are not as simple. Their structure is significantly more convoluted, involving abstract polynomials that do not allow the kind of reasoning that was possible before (standard linear algebra). However, in 2017, Agrawal and Chase introduced a new security notion applicable to pair encodings called the *symbolic property* [AC17]. They also showed how to adapt the previous modular frameworks [AC16, Att16] to define a compiler that takes pair encodings satisfying the symbolic property and produces fully secure predicate encryption schemes under the q-ratio assumption, a new q-type assumption proposed by them that is implied by other assumptions of this kind [LW12]. This symbolic property can be seen as a generalization of the "trick" that we have used in Example 1 to argue the security of the encoding. The main difference is that scalar variables in the PES may be substituted by vectors or matrices (not necessarily scalars as in our example) in such a way that, after the substitution, all the polynomials evaluate to zero, but there is an extra constraint relating the inner product of the vectors that replaced the special variables that guarantees that αs_0 is non-zero. As mentioned by Attrapadung [Att19], the above methodology generalizes the well-known Boneh-Boyen cancellation technique for identity-based encryption [BB11]. What is remarkable about this idea is that the substitution can be used as a "witness" or "certificate" (as coined by the authors of [AC17]) of the security of the scheme. Furthermore, Agrawal and Chase also showed that any pair encoding that is *not trivially broken* satisfies the symbolic property, a result that is closely related to the algebraic characterization of privacy on predicate encodings from [ABS17].

It may seem that after these relevant results on pair encodings, and the similarity with those in the framework of predicate encodings, we are in a position to define a generic negation transformation for pair encodings. However, the

[3] Recall that $\forall \boldsymbol{v} : A\boldsymbol{v} \neq \boldsymbol{z} \Leftrightarrow \exists \boldsymbol{w} : A^\top \boldsymbol{w} = \boldsymbol{0} \wedge \boldsymbol{z}^\top \boldsymbol{w} = 1$ for all compatible A and $\boldsymbol{z}$.

[4] That way, the witness of non-reconstructability can be used as the linear combination for decryption (reconstructability) in the negated encoding and vice versa: the solution for reconstructability can be used as the witness of security in the negated encoding.

more involved structure of pair encodings makes it difficult to find and prove a valid conversion. In fact, recent works have considered the problem of designing such a general negation to be non-trivial (see [Att19, Appendix L.5]), since in the framework of pair encodings it is generally hard to find the mentioned *"certificates"* that can be interpreted as a short proof of security. (Note that any possible NOT transformation would, at least implicitly, use such certificates as decryption credentials for the transformed encoding, whereas the decryption credentials of the original encoding would become the security certificate of the negated one.)

In order to construct a valid negation of pair encodings, we first need to treat them in a simplified manner, closer to linear algebra. To do so, we provide an algebraic characterization of pair encodings (Section 4), whose security can be expressed as a system of matrix equations, very similar to the statement *i)* from Lemma 2. Intuitively, we split the polynomials produced by the encoding into layers, each being a matrix that corresponds to one of the (common, lone or non-lone) variables. We then show how the security of the scheme can be expressed as a linear system involving these matrices. Our characterization makes an structural assumption on the form of the pair encoding (that can be made without loss of generality and has been used in the literature for other purposes [AC17, Att19]). Namely, we assume that EncKey only produces one polynomial that depends on α, which is of the form $\alpha + r_1 b_1$. This assumption introduces a "symmetry" between the nature of key and ciphertext polynomials (now that the special variable α is out of the way) that allows us to express the security of the PES as the symmetric algebraic statement of Definition 3. The next step is to leverage Lemma 1 in order to prove our following lemma, linking the inexistence of a solution to the system in *i)* with the existence of a solution to *ii)*. This is the main tool on which we base our negation transformation. The last (but non-trivial) step is to define a new encoding (in algebraic form) such that the solution from statement *ii)* serves as a decryption credential for it.

Lemma 2. *Let K be a field, let $n \in \mathbb{N}$ and let $\{A_i, B_i, C_i\}_{i\in[n]}$, $\hat{A}, \hat{B}$ be matrices:*

$$A_i \in K^{\ell\times m} \qquad B_i \in K^{r\times s} \qquad C_i \in K^{r\times m}$$
$$\hat{A} \in K^{\ell\times \hat{m}} \qquad \hat{B} \in K^{\hat{r}\times s},$$

for certain $\ell, m, r, s, \hat{m}, \hat{r} \in \mathbb{N}$ and every $i \in [n]$. The following are equivalent:

i) There do not exist X, Y with $X \in K^{r\times\ell}$, $Y \in K^{s\times m}$ such that:

$$\forall i \in [n].\ XA_i + B_iY = C_i \quad \wedge \quad X\hat{A} = 0_{r\times\hat{m}} \quad \wedge \quad \hat{B}Y = 0_{\hat{r}\times m}\ .$$

ii) There exist $Z_1, \ldots, Z_n \in K^{m\times r}$ and $Z_A \in K^{\hat{m}\times r}$, $Z_B \in K^{m\times\hat{r}}$ such that

$$\begin{array}{l} A_1Z_1 + \cdots + A_nZ_n + \hat{A}Z_A = 0_{\ell\times r} \\ \wedge \quad Z_1B_1 + \cdots + Z_nB_n + Z_B\hat{B} = 0_{m\times s} \end{array} \quad \wedge \quad \sum_{i=1}^{n} \mathrm{tr}(C_iZ_i) = 1\ .$$

Proof. Let f be the linear map defined as

$$f : (X, Y) \mapsto (XA_1 + B_1Y,\ \ldots,\ XA_n + B_nY,\ X\hat{A},\ \hat{B}Y)\ .$$

Observe that the first statement of the lemma is equivalent to saying that

$$(C_1, \ldots, C_n, 0_{r\times\hat{m}}, 0_{\hat{r}\times m}) \notin \mathrm{Im}(f)\ ,$$

which, by Lemma 1 is equivalent to the existence of $\varphi : W \to K$, where in this case $W \coloneqq (K^{r\times m})^n \times K^{r\times\hat{m}} \times K^{\hat{r}\times m}$, such that

$$\varphi \circ f = 0 \quad \text{and} \quad \varphi(C_1, \ldots, C_n, 0_{r\times\hat{m}}, 0_{\hat{r}\times m}) = 1\ ,$$

which is equivalent to the existence of matrices $Z_1, \ldots, Z_n \in K^{m\times r}$ and $Z_A \in K^{\hat{m}\times r}$, $Z_B \in K^{m\times\hat{r}}$ such that

$$\forall X, Y.\quad \mathrm{tr}\big(\textstyle\sum_{i=1}^n (XA_i + B_iY)Z_i\big) + \mathrm{tr}\big(X\hat{A}Z_A\big) + \mathrm{tr}\big(\hat{B}YZ_B\big) = 0 \tag{1}$$

$$\text{and}\quad \mathrm{tr}\big(C_1Z_1 + \cdots + C_nZ_n\big) + \mathrm{tr}\big(0_{r\times\hat{m}}Z_A\big) + \mathrm{tr}\big(0_{\hat{r}\times m}Z_B\big) = 1\ , \tag{2}$$

which is equivalent to the second statement of the lemma, *quod erat demonstrandum*. To see why, note that equation (2) is present in both cases and observe that if the second statement of the lemma holds, then (for any X, Y) we have

$$\begin{aligned}
0 &= \mathrm{tr}\big(0_{\ell\times r}\big) + \mathrm{tr}\big(0_{m\times s}\big) \\
&= \mathrm{tr}\big(X\big(A_1Z_1 + \cdots + A_nZ_n + \hat{A}Z_A\big)\big) + \mathrm{tr}\big(\big(Z_1B_1 + \cdots + Z_nB_n + Z_B\hat{B}\big)Y\big) \\
&= \mathrm{tr}\big(\textstyle\sum_{i=1}^n XA_iZ_i\big) + \mathrm{tr}\big(X\hat{A}Z_A\big) + \mathrm{tr}\big(\sum_{i=1}^n Z_iB_iY\big) + \mathrm{tr}\big(Z_B\hat{B}Y\big) \\
&\overset{\dagger}{=} \mathrm{tr}\big(\textstyle\sum_{i=1}^n XA_iZ_i\big) + \mathrm{tr}\big(X\hat{A}Z_A\big) + \mathrm{tr}\big(\sum_{i=1}^n B_iYZ_i\big) + \mathrm{tr}\big(\hat{B}YZ_B\big) \\
&= \mathrm{tr}\big(\textstyle\sum_{i=1}^n (XA_i + B_iY)Z_i\big) + \mathrm{tr}\big(X\hat{A}Z_A\big) + \mathrm{tr}\big(\hat{B}YZ_B\big)\ ,
\end{aligned}$$

where in † we have used the fact that the trace is invariant under cyclic permutations. Finally, to see the converse, note that if equation (1) holds for any X,Y, it must hold for $Y = 0_{s\times m}$, which would imply that for every $X \in K^{r\times\ell}$,

$$\mathrm{tr}\big(X\big(A_1Z_1 + \cdots + A_nZ_n + \hat{A}Z_A\big)\big) = 0\ ,$$

but that can only happen if $A_1Z_1 + \cdots + A_nZ_n + \hat{A}Z_A$ is the zero matrix.

Analogously, evaluating (1) on $X = 0_{r\times\ell}$, we get

$$\mathrm{tr}\big(B_1YZ_1 + \cdots + B_1YZ_n\big) + \mathrm{tr}\big(\hat{B}YZ_B\big) \overset{\dagger}{=} \mathrm{tr}\big(\big(Z_1B_1 + \cdots + Z_nB_n + Z_B\hat{B}\big)Y\big) = 0\ ,$$

for every $Y \in K^{s\times m}$, which can only happen if $Z_1B_1 + \cdots + Z_nB_n + Z_B\hat{B}$ is the null matrix. □

4 Characterization of pair encodings

In this section we propose a characterization of pair encodings that will be used to define our generic transformation for the negated predicate.

The first step towards our characterization is to assume that only one polynomial from EncKey depends on α and is of the form $\alpha + r_1 b_1$. This assumption is without loss of generality[5], and has been utilized before in the literature [AC17, Att19]. The rest of polynomials can be expressed as $\boldsymbol{k} = B_y \boldsymbol{r} + C_y \hat{\boldsymbol{r}}$, for some matrix B_y whose terms are *linear* polynomials in $\mathbb{Z}_N[b_1, \ldots, b_n]$, and some matrix C_y with coefficients in $\mathbb{Z}_N$. Given that $\alpha + r_1 b_1$ is always present, for the sake of notation, we redefine m_3 to be the total number of polynomials produced by KeyGen excluding $\alpha + r_1 b_1$. Similarly, the polynomials from EncCt can be expressed as $\boldsymbol{c} = B'_x \boldsymbol{s} + C'_x \hat{\boldsymbol{s}}$. Such an analogy in the form of $\boldsymbol{k}$ and $\boldsymbol{c}$ (only achieved after getting rid of variable α) allows us to express the encodings in an algebraic form, amenable to be combined with different results of linear algebra.

Definition 3 (Algebraic pair encoding). *An* algebraic *pair encoding scheme for a predicate family $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ indexed by $\kappa = (N, \mathsf{par})$ consists of the following deterministic and efficiently computable algorithms:*

- $\mathsf{Param}^{\mathsf{alg}}(\mathsf{par})$*: on input certain parameters outputs an integer $n \in \mathbb{N}$.*
- $\mathsf{EncKey}^{\mathsf{alg}}(N, x)$*: on input $N \in \mathbb{N}$ and $x \in \mathcal{X}_{(N,\mathsf{par})}$, outputs a list of $n+1$ matrices with coefficients in $\mathbb{Z}_N$, $(B_1, \ldots, B_n, C)$, where B_j has dimension $m_3 \times m_1$, for $j \in [n]$, and C has dimension $m_3 \times m_2$.*
- $\mathsf{EncCt}^{\mathsf{alg}}(N, y)$*: on input $N \in \mathbb{N}$ and $y \in \mathcal{Y}_{(N,\mathsf{par})}$, outputs a list of $n+1$ matrices with coefficients in $\mathbb{Z}_N$, $(B'_1, \ldots, B'_n, C')$, where B'_j has dimension $w_3 \times w_1$, for $j \in [n]$, and C' has dimension $w_3 \times w_2$.*

Furthermore, for every $\kappa = (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$, $P_\kappa(x, y) = 1$ if and only if there exist matrices $E \in \mathbb{Z}_N^{w_1 \times m_3}$ and $E' \in \mathbb{Z}_N^{w_3 \times m_1}$ such that

$$\begin{array}{rlrl} & EB_1 + {B'_1}^\top E' = 1_{w_1 \times m_1} & \wedge & EC = 0_{w_1 \times m_2} \\ \wedge & EB_j + {B'_j}^\top E' = 0_{w_1 \times m_1},\ j \in [2, n] & \wedge & {C'}^\top E' = 0_{w_2 \times m_1} \end{array} \tag{3}$$

where $(B_1, \ldots, B_n, C) \leftarrow \mathsf{EncKey}^{\mathsf{alg}}(N, x)$ and $(B'_1, \ldots, B'_n, C') \leftarrow \mathsf{EncCt}^{\mathsf{alg}}(N, y)$.

Theorem 1 (Characterization). *There exists a pair encoding for predicate family P_κ if and only if there exists an algebraic pair encoding for P_κ. Furthermore, there is an efficient conversion in both directions.*

The above theorem is a consequence of our following two lemmas.

[5] An easy way of arguing that this is w.l.o.g. is to apply the generic dual transformation defined in [AY15] twice. (Note that the dual operation is an involution and a double application of it would preserve the original predicate.)

Lemma 3 (From algebraic to standard). *Let* $(\mathsf{Param}^{\mathsf{alg}}, \mathsf{EncKey}^{\mathsf{alg}}, \mathsf{EncCt}^{\mathsf{alg}})$ *be an algebraic pair encoding scheme for predicate family* $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$. *Then, algorithms* $(\mathsf{Param}, \mathsf{EncKey}, \mathsf{EncCt}, \mathsf{Pair})$ *(defined below) constitute a pair encoding scheme for* P_κ.

- $\mathsf{Param}(\mathsf{par}) :=$ *run* $n \leftarrow \mathsf{Param}^{\mathsf{alg}}(\mathsf{par})$, *output* n *and let* $\boldsymbol{b} = (b_1, \ldots, b_n)$.
- $\mathsf{EncKey}(N, x) :=$ *run* $(B_1, \ldots, B_n, C) \leftarrow \mathsf{EncKey}^{\mathsf{alg}}(N, x)$, *output the vector of polynomials given by* $\alpha + r_1 b_1$ *and* $(b_1 B_1 + \cdots + b_n B_n)\boldsymbol{r} + C\hat{\boldsymbol{r}}$, *where* $\boldsymbol{r} = (r_1, \ldots, r_{m_1})$ *and* $\hat{\boldsymbol{r}} = (\hat{r}_1, \ldots, \hat{r}_{m_2})$.
- $\mathsf{EncCt}(N, y) :=$ *run* $(B'_1, \ldots, B'_n, C') \leftarrow \mathsf{EncCt}^{\mathsf{alg}}(N, y)$, *output the vector of polynomials given by* $(b_1 B'_1 + \cdots + b_n B'_n)\boldsymbol{s} + C'\hat{\boldsymbol{s}}$, *where* $\boldsymbol{s} = (s_0, \ldots, s_{w_1 - 1})$ *and* $\hat{\boldsymbol{s}} = (\hat{s}_1, \ldots, \hat{s}_{w_2})$.
- $\mathsf{Pair}(N, x, y) :=$ *find matrices* (E, E') *satisfying equation (3), that exist if and only if* $P_\kappa(x, y) = 1$, *output* $\left(\left(\mathbf{1}_{w_1} \; -E\right), -E'\right)$.

Proof. Observe that the structural constraints on the polynomials of EncKey and EncCt are satisfied. To see reconstructability, simply note that for any $N \in \mathbb{N}$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ with $P(x, y) = 1$, and for (E, E') satisfying (3), it holds:

$$\boldsymbol{s}^\top \left(\mathbf{1}_{w_1} \; -E\right) \begin{pmatrix} \alpha + r_1 b_1 \\ (b_1 B_1 + \cdots + b_n B_n)\boldsymbol{r} + C\hat{\boldsymbol{r}} \end{pmatrix} - \left(\boldsymbol{s}^\top \left(b_1 {B'_1}^\top + \cdots + b_n {B'_n}^\top\right) + \hat{\boldsymbol{s}}^\top {C'}^\top\right) E'\boldsymbol{r}$$
$$= s_0(\alpha + r_1 b_1) - \boldsymbol{s} b_1 \left(1_{w_1 \times m_1}\right) \boldsymbol{r} = s_0 \alpha.$$

For security, note that if the new pair encoding were trivially broken, there would exist a pair $(x, y) \in \mathcal{X}_\kappa \times \mathcal{Y}_\kappa$ with $P_\kappa(x, y) = 0$, and matrices E, E' satisfying equation (3). For details about this fact, we refer to the proof Lemma 4 (the part about reconstructability). □

Lemma 4 (From standard to algebraic). *Let* $(\mathsf{Param}, \mathsf{EncKey}, \mathsf{EncCt}, \mathsf{Pair})$ *be a pair encoding scheme*[6] *for predicate family* $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$. *Then, algorithms* $(\mathsf{Param}^{\mathsf{alg}}, \mathsf{EncKey}^{\mathsf{alg}}, \mathsf{EncCt}^{\mathsf{alg}})$ *(defined below) constitute an algebraic pair encoding scheme for* P_κ.

- $\mathsf{Param}^{\mathsf{alg}}(\mathsf{par}) := \mathsf{Param}(\mathsf{par})$.
- $\mathsf{EncKey}^{\mathsf{alg}}(N, x) :=$ *run* $(\alpha + r_1 b_1, \boldsymbol{k}) \leftarrow \mathsf{EncKey}(N, x)$, *and let* $m_3 = |\boldsymbol{k}|$. *For* $j \in [n]$, *define matrix* B_j *as the matrix whose element at the* ℓ*-th row and* i*-th column is the coefficient of monomial* $r_i b_j$ *in polynomial* k_ℓ. *Define* C *as the matrix whose element at the* ℓ*-th row and* i'*-th column is the coefficient of monomial* $\hat{r}_{i'}$ *in polynomial* k_ℓ, *for* $i \in [m_1]$, $i' \in [m_2]$ *and* $\ell \in [m_3]$. *Output* $(B_1, \ldots, B_n, C)$.
- $\mathsf{EncCt}^{\mathsf{alg}}(N, y) :=$ *run* $\boldsymbol{c} \leftarrow \mathsf{EncCt}(N, y)$. *For* $j \in [n]$, *define matrix* B'_j *as the matrix whose element at the* ℓ*-th row and* $(i{+}1)$*-th column is the coefficient*

[6] Recall that we are assuming, without loss of generality, that the first polynomial produced by EncKey is $\alpha + r_1 b_1$ and that α does not appear anywhere else.

of monomial $s_i b_j$ in polynomial c_ℓ. Define C' as the matrix whose element at the ℓ-th row and i'-th column is the coefficient of monomial $\hat{s}_{i'}$ in polynomial c_ℓ, for $i \in [0, w_1-1]$, $i' \in [w_2]$ and $\ell \in [w_3]$. Output $(B'_1, \ldots, B'_n, C')$.

Proof. Note that the structural constraints on the PES enforce that for every $N \in \mathbb{N}$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$, $(\alpha + r_1 b_1, \boldsymbol{k}) \leftarrow \mathsf{EncKey}(N, x)$, $\boldsymbol{c} \leftarrow \mathsf{EncCt}(N, y)$, $(B_1, \ldots, B_n, C) \leftarrow \mathsf{EncKey}^{\mathsf{alg}}(N, x)$, $(B'_1, \ldots, B'_n, C') \leftarrow \mathsf{EncCt}^{\mathsf{alg}}(N, y)$, it holds:

$$\boldsymbol{k} = (b_1 B_1 + \cdots + b_n B_n)\boldsymbol{r} + C\hat{\boldsymbol{r}} \quad \text{and} \quad \boldsymbol{c} = (b_1 B'_1 + \cdots + b_n B'_n)\boldsymbol{s} + C'\hat{\boldsymbol{s}} \ .$$

Now, note that, due to reconstructability of the original encoding, for any $N \in \mathbb{N}$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $P(x, y) = 1$, if we let $\big((\boldsymbol{v}\ E), E'\big) \leftarrow \mathsf{Pair}(N, x, y)$, it holds:

$$\boldsymbol{s}^\top (\boldsymbol{v}\ E) \begin{pmatrix} \alpha + r_1 b_1 \\ \boldsymbol{k} \end{pmatrix} + \boldsymbol{c}^\top E' \boldsymbol{r} = \alpha s_0 \ ,$$

which is equivalent to $\boldsymbol{s}^\top E \boldsymbol{k} + \boldsymbol{c}^\top E' \boldsymbol{r} = -s_0 r_1 b_1 \wedge \boldsymbol{v} = \boldsymbol{1}_{w_1}$, but then:

$$\boldsymbol{s}^\top E\big((b_1 B_1 + \cdots + b_n B_n)\boldsymbol{r} + C\hat{\boldsymbol{r}}\big) + \big(\boldsymbol{s}^\top (b_1 B_1'^\top + \cdots + b_n B_n'^\top) + \hat{\boldsymbol{s}}^\top C'^\top\big) E' \boldsymbol{r} = -s_0 r_1 b_1$$

and because the above equality must hold ***symbolically***, it must be the case that $EB_1 + B'_1 E' = 1_{w_1 \times m_1}$ and $EB_j + B'_j E' = 0_{w_1 \times m_1}$ for every $j \in [2, n]$. Moreover, $EC = 0_{w_1 \times m_2}$ and $C'^\top E' = 0_{w_2 \times m_1}$. Finally, note that the non-reconstructability of the original encoding enforces that the above system does not have a solution when $P_\kappa(x, y) = 0$. □

5 Generic negation of algebraic pair encodings

Although the general definition of pair encodings defines polynomials with coefficients over $\mathbb{Z}_N$ for an arbitrary integer $N \in \mathbb{N}$. In this section we assume that N is a prime number and write p instead. The reason is that our transformation for the negated encoding leverages a result from linear algebra (our Lemma 2) which requires that the underlying structure be a field. Note that this restriction does not significantly weaken our result, since prime-order groups are preferred over composite over groups.

Theorem 2. *Let $(\mathsf{Param}^{\mathsf{alg}}, \mathsf{EncKey}^{\mathsf{alg}}, \mathsf{EncCt}^{\mathsf{alg}})$ be an algebraic pair encoding for a predicate family $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0, 1\}$. The encoding $(\overline{\mathsf{Pair}}, \overline{\mathsf{EncKey}}, \overline{\mathsf{EncCt}})$ described in Figure 1 is an algebraic pair encoding for the predicate family $\overline{P}_\kappa$ given by $\overline{P}(x, y) = 1 \Leftrightarrow P(x, y) = 0$ for all $x \in \mathcal{X}_\kappa$, $y \in \mathcal{Y}_\kappa$.*

Proof. We need to show that whenever $P(x, y) = 0$, there exist matrices $\overline{E}$ and $\overline{E}'$ of dimension $w_1 \times (1 + m_1 n + m_2)$ and $(w_1 + w_1 n + w_2) \times m_1$ respectively, with coefficients in $\mathbb{Z}_p$, such that:

$$\begin{array}{llll} & \overline{E}\overline{B}_0 + \overline{B}_0'^\top \overline{E}' = 1_{w_1 \times m_1} & \wedge & \overline{E}\overline{C} = 0_{w_1 \times m_3} \\ \wedge & \overline{E}\overline{B}_j + \overline{B}_j'^\top \overline{E}' = 0_{w_1 \times m_1},\ j \in [n+1] & \wedge & \overline{C}'^\top \overline{E}' = 0_{w_3 \times m_1} \ , \end{array} \qquad (4)$$

Let $(\mathsf{Param}^{\mathsf{alg}}, \mathsf{EncKey}^{\mathsf{alg}}, \mathsf{EncCt}^{\mathsf{alg}})$ be an algebraic pair encoding scheme. We define the following (algebraic) PES:

- $\overline{\mathsf{Param}}(\mathsf{par}) \coloneqq$ run $n \leftarrow \mathsf{Param}^{\mathsf{alg}}(\mathsf{par})$, output $n{+}2$.
- $\overline{\mathsf{EncKey}}(p, x) \coloneqq$ run $(B_1, \ldots, B_n, C) \leftarrow \mathsf{EncKey}^{\mathsf{alg}}(p, x)$, and let:

$$\overline{B}_0 \coloneqq \begin{pmatrix} \mathbf{1}_{m_1}^\top \\ \hline 1_{m_1 n \times m_1} \\ \hline 0_{m_2 \times m_1} \end{pmatrix} \quad \overline{B}_i \coloneqq \begin{pmatrix} \mathbf{0}_{m_1}^\top \\ \hline \boldsymbol{e}_i^n \otimes I_{m_1} \\ \hline 0_{m_2 \times m_1} \end{pmatrix} \quad \overline{B}_{n+1} \coloneqq \begin{pmatrix} \mathbf{1}_{m_1}^\top \\ \hline 0_{m_1 n \times m_1} \\ \hline 0_{m_2 \times m_1} \end{pmatrix} \quad \overline{C} \coloneqq \begin{pmatrix} \mathbf{0}_{m_3}^\top \\ \hline B_1^\top \\ \vdots \\ B_n^\top \\ \hline C^\top \end{pmatrix} .$$

  Output $(\overline{B}_0, \overline{B}_1, \ldots, \overline{B}_{n+1}, \overline{C})$.

- $\overline{\mathsf{EncCt}}(p, y) \coloneqq$ run $(B'_1, \ldots, B'_n, C') \leftarrow \mathsf{EncCt}^{\mathsf{alg}}(p, y)$, let $\overline{B}'_0$ be the zero matrix of $w_1(1{+}n){+}w_2$ rows and w_1 columns and let:

$$\overline{B}'_i \coloneqq \begin{pmatrix} 0_{w_1 \times w_1} \\ \hline -\boldsymbol{e}_i^n \otimes I_{w_1} \\ \hline 0_{w_2 \times w_1} \end{pmatrix} \quad \overline{B}'_{n+1} \coloneqq \begin{pmatrix} I_{w_1} - 1_{w_1 \times w_1} \\ \hline 0_{w_1 n \times w_1} \\ \hline 0_{w_2 \times w_1} \end{pmatrix} \quad \overline{C}' \coloneqq \begin{pmatrix} 0_{w_1 \times w_3} \\ \hline {B'_1}^\top \\ \vdots \\ {B'_n}^\top \\ \hline {C'}^\top \end{pmatrix} .$$

  Output $(\overline{B}'_0, \overline{B}'_1, \ldots, \overline{B}'_{n+1}, \overline{C}')$.

Fig. 1. Generic negation of algebraic pair encoding schemes.

where $(\overline{B}_0, \ldots, \overline{B}_{n+1}, \overline{C}) \leftarrow \overline{\mathsf{EncKey}}(p, x)$, $(\overline{B}'_0, \ldots, \overline{B}'_{n+1}, \overline{C}') \leftarrow \overline{\mathsf{EncCt}}(p, y)$. Now, our original encoding guarantees that $P(x, y) = 0$ if and only if there **do not** exist matrices E, E' such that:

$$\begin{array}{rlrl} & EB_1 + {B'_1}^\top E' = 1_{w_1 \times m_1} & \wedge & EC = 0_{w_1 \times m_2} \\ \wedge & EB_j + {B'_j}^\top E' = 0_{w_1 \times m_1},\ j \in [2, n] & \wedge & C'^\top E' = 0_{w_2 \times m_1} \ , \end{array}$$

for $(B_1, \ldots, B_n, C) \leftarrow \mathsf{EncKey}(p, x)$ and $(B'_1, \ldots, B'_n, C') \leftarrow \mathsf{EncCt}(p, y)$. But that is equivalent, in virtue of Lemma 2, to the existence of $Z_1, \ldots, Z_n \in \mathbb{Z}_p^{m_1 \times w_1}$, $Z_A \in \mathbb{Z}_p^{m_2 \times w_1}$ and $Z_B \in \mathbb{Z}_p^{m_1 \times w_2}$ such that:[7]

$$\begin{array}{rlrl} & B_1 Z_1 + \cdots + B_n Z_n + C Z_A = 0_{m_3 \times w_1} & & \\ \wedge & Z_1 {B'_1}^\top + \cdots + Z_n {B'_n}^\top + Z_B C'^\top = 0_{m_1 \times w_3} & \wedge & \mathrm{tr}(1_{w_1 \times m_1} Z_1) = 1 \ . \end{array} \tag{5}$$

[7] To see why, set the matrices in Lemma 2 to $A_i \coloneqq B_i$, $B_i \coloneqq {B'_i}^\top$, for $i \in [n]$ and $C_1 \coloneqq 1_{w_1 \times m_1}$, $C_j \coloneqq 0_{w_1 \times m_1}$ for $j \in [2, n]$. Also, $\hat{A} \coloneqq C$ and $\hat{B} \coloneqq C'^\top$.

Now, for certain $\boldsymbol{v} \in \mathbb{Z}_p^{w_1}$ and $V \in \mathbb{Z}_p^{m_1 \times w_1}$ we can consider the matrices:

$$\overline{E} \coloneqq \left(\boldsymbol{v} \,|\, Z_1^\top \ldots Z_n^\top \,|\, Z_A^\top\right) \quad \text{and} \quad \overline{E}' \coloneqq \left(V \,|\, Z_1 \ldots\ Z_n \,|\, Z_B\right)^\top \ , \tag{6}$$

and observe that they satisfy all the equations in (4) if we set $\boldsymbol{v}$ to be the first column of $Z_1^\top$ multiplied by -1 (with the exception that $v_1 = 0$) and we set V to be the null matrix except for its first row, that is set to $-\boldsymbol{v}^\top$.

To conclude, observe that the converse is also true, i.e., if the equations in (4) admit a solution, then (5) is satisfiable. To see this, note that the left-hand side equations of (4) imply that any solution to them must be of the form of (6) for certain $\boldsymbol{v}$, V, $Z_1, \ldots, Z_n, Z_A, Z_B$. Furthermore, the right-hand side equations of (4) guarantee that such matrices Z_i, for $i \in \{1, \ldots, n, A, B\}$ satisfy (5). Therefore, we have shown that $P(x, y) = 0$ iff the equations in (4) have a solution. □

Observe that, in general, if $(m_1, m_2, m_3, w_1, w_2, w_3, n)$ are the parameters of the original encoding, our negated transformation will produce an encoding with parameters $\overline{n} = n+2$ and:

$$\begin{array}{lll} \overline{m}_1 = m_1 & \overline{m}_2 = m_3 & \overline{m}_3 = 1 + m_1 n + m_2 \\ \overline{w}_1 = w_1 & \overline{w}_2 = w_3 & \overline{w}_3 = w_1(1+n) + w_2 - 1 \ . \end{array}$$

Note that, although the negated encoding may seem to have a much larger size compared to the original one, the matrices associated to the new encoding are actually very sparse and thus, our transformation will barely impact the performance of the ABE scheme build from the negated encoding.

Furthermore, note that our generic negation is compatible with the promising dynamic pair encoding composition technique very recently proposed by Attrapadung [Att19]. We believe our new transformation complements his work which could only achieve non-monotone formulae composition in a semi-generic (but dynamic) manner, since the composition had to rely on encodings for which a negated version was available.

6 Consequences of our results

Since Attrapadung introduced the notion of pair encoding schemes and the modular framework for constructing fully secure ABE from them [Att14], there have been several works [AC17, AC16, Att19] refining this framework and proposing new encoding schemes for different predicates, that sometimes enjoy extra properties (e.g., constant ciphertext size). The community has made a significant effort on building the negated version of most of the encodings from the literature, which in some cases is significantly more involved. However, there are still encodings for which not negation is known. Our generic transformation puts an end to this situation, since we can now take any encoding and immediately obtain its negated counterpart. A relevant example of a PES with (previously) unknown negation is the case of doubly spatial encryption.

6.1 PES for negated doubly spatial encryption

Doubly spatial encryption [Ham11] is an important primitive that generalizes both spatial encryption[8] [BH08] and *negated spatial encryption*, defined by Attrapadung and Libert [AL10]. It can be used to capture complex predicates and build flexible revocation systems. Its relevance is evidenced by the fact that a variant of it, called key-policy over doubly spatial encryption (defined by Attrapadung [Att14]), generalizes KP-ABE and leads to efficient unbounded KP-ABE schemes with large universes and KP-ABE with short ciphertexts. Given a field K, the doubly spatial predicate, over sets $\mathcal{X} := K^d \times K^{d\times\ell}$ and $\mathcal{Y} := K^d \times K^{d\times\ell'}$, $P((\boldsymbol{x}, X), (\boldsymbol{y}, Y))$, is defined as 1 if and only if the affine spaces $\boldsymbol{x} + \mathrm{span}(X)$ and $\boldsymbol{y} + \mathrm{span}(Y)$ intersect.

In the same way that negated spatial encryption generalizes spatial encryption and serves as its revocation analogue, unifying existing primitives (for example, it subsumes non-zero-mode IPE), negated *doubly* spatial encryption is a more expressive and very powerful primitive that deserves our attention. However, to the best of our knowledge, there does not exist a general pair encoding scheme for negated doubly spatial encryption in the literature. Attrapadung [Att14] provided a pair encoding for doubly spatial encryption and a negated version, for which he had to restrict one of the attributes (originally the ciphertext attribute) to be confined to just a vector instead of a general affine space. This encoding gave birth to the first fully-secure negated spatial encryption scheme, but it is not the negated version of *doubly* spatial encryption. In the rest of this section, we describe how to obtain the first, to the best of our knowledge, pair encoding scheme for negated doubly spatial encryption without restrictions.

We start from the following PES for doubly spatial encryption (over $\mathbb{Z}_N$) from [Att14]. (With $m_1 = 1, m_2 = 0, m_3 = \ell'+1$ and $w_1 = 1, w_2 = 0, w_3 = \ell+1$.)

$$\begin{aligned}
\mathsf{Param}(\mathsf{par}) &\to d+1 \text{ and let } \boldsymbol{b} = (b_0, \boldsymbol{b}') = (b_0, b_1, \ldots, b_d)\\
\mathsf{EncKey}(N, (\boldsymbol{y}, Y)) &:= \{\alpha + r_1 b_0 + r_1 \boldsymbol{y}^\top \boldsymbol{b}',\ r_1 Y^\top \boldsymbol{b}'\}\\
\mathsf{EncCt}(N, (\boldsymbol{x}, X) &:= \{\quad s_0 b_0 + s_0 \boldsymbol{x}^\top \boldsymbol{b}',\ s_0 X^\top \boldsymbol{b}'\}\ .
\end{aligned}$$

We refer to [Att14] for a proof of security and reconstructability.

In order to apply our negated transformation to this encoding, we first need to modify it so that it satisfies our structural assumption (see the first paragraph of our Section 4). For this, we can apply the conversion defined by Attrapadung [Att19, Section 4]. If we do so, we will get and encoding with $m_1 = 2$, $m_2 = 0$, $m_3 = \ell'+1$ and $w_1 = 2$, $w_2 = 0$, $w_3 = \ell+2$ that looks as follows (after renaming some variables):

[8] Spatial encryption is already a quite powerful predicate, that generalizes hierarchical identity-based encryption (HIBE).

PES for predicate $P((\boldsymbol{x}, X), (\boldsymbol{y}, Y)) = 1 \Leftrightarrow (\boldsymbol{x} + \mathrm{span}(X)) \cap (\boldsymbol{y} + \mathrm{span}(Y)) = \emptyset$.

Where $\mathcal{X} := \mathbb{Z}_N^d \times \mathbb{Z}_N^{d \times \ell}$ and $\mathcal{Y} := \mathbb{Z}_N^d \times \mathbb{Z}_N^{d \times \ell'}$ for integers N, d, ℓ, ℓ'.

- $\mathsf{Param}(\mathsf{par}) \to d+4$ and let $\boldsymbol{b} = (b_0, \boldsymbol{b}', t, u, v)$ with $\boldsymbol{b}' = (b_1, \ldots, b_d)$.
- $\mathsf{EncKey}(N, (\boldsymbol{y}, Y)) :=$
  $\left\{ r_1(b_0 + t),\ r_2 u + \hat{r}_1,\ r_1 v + \hat{r}_1,\ (Y_j \hat{\boldsymbol{r}}' + r_2 b_j + \hat{r}_1 y_j)_{j \in [d]} \right\}$ (also $\alpha + r_1 b_0$).
- $\mathsf{EncCt}(N, (\boldsymbol{x}, X)) :=$
  $\left\{ s_0 t - \hat{s}_1,\ s_1 v - \hat{s}_1,\ s_1 u - \hat{s}_2,\ (X_j \hat{\boldsymbol{s}}' - s_1 b_j + \hat{s}_2 x_j)_{j \in [d]} \right\}$.

Here, $\boldsymbol{r} := (r_1, r_2)$, $\hat{\boldsymbol{r}} := (\hat{r}_1, \hat{\boldsymbol{r}}')$ with $\hat{\boldsymbol{r}}' := (\hat{r}_2, \ldots, \hat{r}_{\ell'+1})$, and $\boldsymbol{s} := (s_0, s_1)$, $\hat{\boldsymbol{s}} := (\hat{s}_1, \hat{s}_2, \hat{\boldsymbol{s}}')$ with $\hat{\boldsymbol{s}}' := (\hat{s}_3, \ldots, \hat{s}_{\ell+2})$.

Fig. 2. Simplified PES for negated doubly spatial encryption.

$$\begin{aligned}
\mathsf{Param}(\mathsf{par}) &\to d+3 \text{ and let } \boldsymbol{b} = (b_0, \boldsymbol{b}', b_{d+1}, b_{d+2}) \text{ with } \boldsymbol{b}' = (b_1, \ldots, b_d) \\
\mathsf{EncKey}(N, (\boldsymbol{y}, Y)) &:= \{r_1 b_{d+2} + r_2 b_{d+1} + r_2 \boldsymbol{y}^\top \boldsymbol{b}', \quad r_2 Y^\top \boldsymbol{b}'\} \text{ (also } \alpha + r_1 b_0) \\
\mathsf{EncCt}(N, (\boldsymbol{x}, X) &:= \{s_0 b_0 + s_1 b_{d+2}, \quad s_1 b_{d+1} + s_1 \boldsymbol{x}^\top \boldsymbol{b}', \quad s_1 X^\top \boldsymbol{b}'\} \ .
\end{aligned}$$

Applying our negation transformation to the above encoding, we obtain the pair encoding described in Figure 3 (presented in Appendix A), where we have renamed[9] some common variables for the sake of readability. In Appendix A.1 we show how we can slightly simplify the encoding from Figure 3 and derive the encoding that we present in Figure 2. Our Theorem 2 guarantees that it is a valid encoding for the negated doubly spatial encryption predicate, but we provide an independent proof in Appendix A.2.

The process of applying our generic negation by hand may seem tedious (but it seems necessary if we want to give an explicit description of an encoding that is parametric in size, like the one for negated doubly spatial encryption). However, notice that this process can be easily delegated to a computer, which does not need to have an explicit definition of the negated encoding. Instead, it can start from the non-negated encoding and apply the negation on the fly.

6.2 Other implications of our transformation

Expressivity of pair encoding schemes. A very important and long-standing open question about pair encoding schemes is *how expressive they really are*. They have led to breakthrough constructions such as constant-size ciphertext KP-ABE (with large universes) [Att14], fully-secure functional encryption for regular languages [Att14], completely-unbounded KP-ABE for non-monotone span

[9] Before applying the transformation, we rename $b_0 \mapsto t$, $b_{d+1} \mapsto u$, $b_{d+2} \mapsto v$. After the transformation, the two new common variables are named b_0 and w respectively.

programs (NSP) over large universes [Att19]. However, it is still unknown where their limit is. We believe our results bring new insight to answer this question and improve our understanding of pair encodings and their expressivity.

For example, there exist pair encodings for regular languages, where key attributes represent deterministic finite-state automata (DFSA), ciphertext attributes represent (arbitrarily long) words, and the predicate is defined as 1 iff the automaton accepts the word. However, building ABE for context-free languages (CFL) from pairings is still an important open problem, so it would be desirable to understand whether CFL can be constructed from pair encoding schemes. Our results imply that:

The set of predicates that can be expressed with PES is closed under negation.

This tells us new non-trivial information about what predicates can be expressed with a PES. In particular, it suggests that building PES for context-free languages may be harder than we think or even impossible. Note that context-free languages are not closed under complementation [HU79] and, consequently, if we can build a PES for CFL, we could build a PES for a predicate class that is strictly more powerful than CFL (at least the union of CFL and coCFL[10]). Of course, this reasoning does not allow us to roundly conclude anything, but it serves as an evidence of the difficulty of this problem.

Potential performance improvements. Not only does our generic transformation broaden the class of predicates that can be captured by pair encoding schemes, but it also can lead to efficiency improvements in actual ABE constructions. Observe the peculiar structure of the negated encodings produced with our transformation from Figure 1. All of the matrices associated to common variables, $\overline{B}_i$ and $\overline{B}'_i$, have a fixed structure that is independent of the key attribute and the ciphertext attribute respectively (only the part associated to lone variables is dependent on the attributes). Furthermore, observe that they are arguably sparse. We can conclude that all pair encoding schemes admit a representation (an encoding for the same predicate) in this form, since we can always apply our transformation twice, leveraging the fact that the negation is an involution. However, in many cases it may be simpler to arrive at the mentioned structure more directly, by simply applying linear combinations and variable substitutions. What is important is that such a representation always exists.

This observation opens the possibility of splitting the computation of ciphertexts and secret keys into an offline part (before the attribute value is known) and an online part (once the attribute has been determined). Observe that such a strategy can bring significant performance improvements, given that operations involving common variables require a group exponentiation per matrix coefficient (since the common variables are available in the master public key in the form of group elements, with unknown discrete logarithm)[11], whereas operations involving lone variables can be batched together, reducing the number of exponentiations (one can do linear algebra over the field $\mathbb{Z}_N$ and perform one single

[10] We denote by coCFL the class of languages whose complement is context-free.
[11] See the ABE compiler from PES described in Appendix B.2.

exponentiation at the end). This is because the value of lone variables is freshly sampled during the computation and, therefore, known. This approach would not only reduce the online encryption and key generation time, but also the total time, since the offline computation can be reused for different attributes after it has been computed once.

7 Conclusions and future work

Pair encodings are a simple, yet powerful, tool for building complex fully secure attribute-based encryption schemes. In this work, we have presented a generic transformation that takes any pair encoding scheme and negates its predicate. This construction finally solves a problem that was open since 2015 [AY15] and that has been considered to be non-obvious by several recent works [AC17, Att19]. Along the way, we have defined new results that improve our understanding of pair encodings and can be of independent interest, including a new encoding (previously unknown) for negated doubly spatial encryption, obtained with our transformation.

We propose several directions for future work. On the theoretical side, it would be interesting to explore whether our negation transformation can lead to simpler encodings as in [ABS17]. In their work, Ambrona *et al.* show how, applying their negation to an encoding for monotone span programs [KW93] and after performing some simplifications, the new encoding is more compact and leads to an ABE that is twice as fast as the original one. The fact that the encoding is negated does not spoil its usage, since span programs are closed under negation and can be tweaked to implement the original functionality. The same technique of negating the encoding also results into a successful simplification in the case of arithmetic span programs. We believe the same kind of phenomenon can occur when negating pair encodings with our technique, potentially producing simpler encodings.

A very recent work [AT20] provides a new framework for constructing ABE schemes that support unbounded and dynamic predicate compositions whose security is proven under the standard matrix Diffie-Hellman assumption (generalizing the result by Attrapadung [Att19], which achieved the same kind of composition under the q-ratio assumption). The work by Attrapadung and Tomida [AT20] enables generic conjunctive and disjunctive compositions (which lead to *monotone* Boolean formula compositions). Extending their techniques in order to design a generic negation under standard assumptions is a very appealing direction for future work. (Note that the negation that we have provided in this work is applicable to the framework of Agrawal and Chase [AC17], thus it also relies on the less standard q-ratio assumption.)

On the practical side, it would be interesting to implement and evaluate the performance improvements that we propose in Section 6.2, exploiting the singular structure of the encodings produced by our transformation.

Acknowledgments

I would like to express my sincere gratitude to Nuttapong Attrapadung, for very fruitful discussions at the early stages of this project, and for pointing out that the transformation described herein would lead to an encoding for negated doubly spatial encryption. I would also like to thank Mehdi Tibouchi, for his help with the formulation of Lemma 1, which led to a simple proof for Lemma 2; and Elena Gutiérrez, for her advice on automata theory, and for all her feedback. Finally, I would like to thank the anonymous reviewers of PKC 2021, for their valuable time and multiple suggestions.

References

ABS17. Miguel Ambrona, Gilles Barthe, and Benedikt Schmidt. Generic transformations of predicate encodings: Constructions and applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 36–66. Springer, Heidelberg, August 2017.

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.

AC17. Shashank Agrawal and Melissa Chase. Simplifying design and analysis of complex predicate encryption schemes. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 627–656. Springer, Heidelberg, April / May 2017.

AI09. Nuttapong Attrapadung and Hideki Imai. Dual-policy attribute based encryption. In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *ACNS 09*, volume 5536 of *LNCS*, pages 168–185. Springer, Heidelberg, June 2009.

AL10. Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 384–402. Springer, Heidelberg, May 2010.

AT20. Nuttapong Attrapadung and Junichi Tomida. Unbounded dynamic predicate compositions in ABE from standard assumptions. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 405–436. Springer, Heidelberg, December 2020.

Att14. Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014.

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.

Att19. Nuttapong Attrapadung. Unbounded dynamic predicate compositions in attribute-based encryption. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 34–67. Springer, Heidelberg, May 2019.

AY15. Nuttapong Attrapadung and Shota Yamada. Duality in ABE: Converting attribute based encryption for dual predicate and dual policy via computational encodings. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 87–105. Springer, Heidelberg, April 2015.

BB11. Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, October 2011.

Bei11. Amos Beimel. *Secret-Sharing Schemes: A Survey*, pages 11–46. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

BH08. Dan Boneh and Michael Hamburg. Generalized identity based and broadcast encryption schemes. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 455–470. Springer, Heidelberg, December 2008.

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.

CMP17. Sanjit Chatterjee, Sayantan Mukherjee, and Tapas Pandit. CCA-secure predicate encryption from pair encoding in prime order groups: Generic and efficient. In Arpita Patra and Nigel P. Smart, editors, *INDOCRYPT 2017*, volume 10698 of *LNCS*, pages 85–106. Springer, Heidelberg, December 2017.

CW13. Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460. Springer, Heidelberg, August 2013.

CW14. Jie Chen and Hoeteck Wee. Dual system groups and its applications — compact HIBE and more. Cryptology ePrint Archive, Report 2014/265, 2014. http://eprint.iacr.org/2014/265.

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 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

GVW13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

Ham11. Mike Hamburg. Spatial encryption. Cryptology ePrint Archive, Report 2011/389, 2011. http://eprint.iacr.org/2011/389.

HU79. John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

KSGA16. Jongkil Kim, Willy Susilo, Fuchun Guo, and Man Ho Au. A tag based encoding: An efficient encoding for predicate encryption in prime order groups. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 3–22. Springer, Heidelberg, August / September 2016.

KSW08. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P.

Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, April 2008.

KW93. Mauricio Karchmer and Avi Wigderson. On span programs. In *Proceedings of Structures in Complexity Theory*, pages 102–111, 1993.

LW10. Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, Heidelberg, February 2010.

LW11. Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Heidelberg, May 2011.

LW12. Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 180–198. Springer, Heidelberg, August 2012.

OSW07. Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 2007*, pages 195–203. ACM Press, October 2007.

RW13. Yannis Rouselakis and Brent Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 463–474. ACM Press, November 2013.

Sha84. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, August 1984.

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.

Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, August 2009.

Wat12. Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 218–235. Springer, Heidelberg, August 2012.

Wee14. Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Heidelberg, February 2014.

A Pair encoding for negated doubly spatial encryption

A.1 Building the encoding

A direct application of our negated transformation (Figure 1) to the encoding for doubly spatial encryption from [Att14] (after minor modifications so that it satisfies our structural constraints) leads to the encoding from Figure 3. This encoding can be simplified, as the following reasoning shows that not all the polynomials are needed for reconstructability.

The only way to get polynomial $s_0 r_1 b_0$ (and consequently αs_0) as a linear combination of polynomials from $L = \boldsymbol{s} \otimes \boldsymbol{k} \cup \boldsymbol{c} \otimes \boldsymbol{r}$ is through the two first polynomials in the key (multiplied by s_0): $s_0 r_1 b_0 + s_0 r_1 w$ and $s_0 r_1 b_0 + s_0 r_1 t$. For that, we need to express monomial $s_0 r_1 w$ or monomial $s_0 r_1 t$ as a linear combination of other polynomials in L. The former is impossible to obtain (since monomial $s_0 r_1 w$ does not appear in any other polynomial in L). The latter can be achieved only through polynomial $r_1 s_0 t - r_1 \hat{s}_1 \in L$. Again, that requires to arrive at polynomial $r_1 \hat{s}_1$, which is present only in $r_1 s_1 v - r_1 \hat{s}_1$. Furthermore, $r_1 s_1 v$ can only be (additionally) found in $s_1 r_1 v + s_1 \hat{r}_1$. However, $s_1 \hat{r}_1$ is present in several polynomials in L, namely: $s_1 r_2 u + s_1 \hat{r}_1$ and $s_1 (Y_j \hat{\boldsymbol{r}}' + r_2 b_j + \hat{r}_1 y_j)_{j \in [d]}$. The former contains a monomial, $s_1 r_2 u$, that only additionally appears in $r_2 s_1 u - r_2 \hat{s}_2$, but $r_2 \hat{s}_2$ is only present in polynomials $r_2 (X_j \hat{\boldsymbol{s}}' - s_1 b_j + \hat{s}_2 x_j)_{j \in [d]}$.

Consequently, reconstructability will be possible if there exist coefficients β_j and γ_j for all $j \in [0, d]$ such that:

$$\begin{aligned} s_1 \hat{r}_1 \;=\; & \beta_0 (s_1 r_2 u + s_1 \hat{r}_1) + \textstyle\sum_{j \in [d]} \beta_j s_1 (Y_j \hat{\boldsymbol{r}}' + r_2 b_j + \hat{r}_1 y_j) \\ & + \gamma_0 (r_2 s_1 u - r_2 \hat{s}_2) + \textstyle\sum_{j \in [d]} \gamma_j r_2 (X_j \hat{\boldsymbol{s}}' - s_1 b_j + \hat{s}_2 x_j) \;\; . \end{aligned}$$

Considering the different monomials in both sides of the equation, we deduce:

$$\begin{array}{rlrl} s_1 \hat{r}_1 : & 1 = \beta_0 + \sum_{j \in [d]} \beta_j y_j & r_2 \hat{s}_2 : & 0 = -\gamma_0 + \sum_{j \in [d]} \gamma_j x_j \\ s_1 r_2 u : & 0 = \beta_0 + \gamma_0 & s_1 r_2 b_j : & 0 = \beta_j - \gamma_j \quad \forall j \in [d] \\ s_1 \hat{\boldsymbol{r}}' : & \mathbf{0}_{\ell'} = \sum_{j \in [d]} \beta_j Y_j & r_2 \hat{\boldsymbol{s}}' : & \mathbf{0}_{\ell} = \sum_{j \in [d]} \gamma_j X_j \end{array}$$

Consequently, reconstructability is possible if there exist coefficients β_j for all $j \in [d]$ such that:

$$1 = \textstyle\sum_{j \in [d]} \beta_j (y_j - x_j) \quad \wedge \quad \mathbf{0}_{\ell'} = \sum_{j \in [d]} \beta_j Y_j \quad \wedge \quad \mathbf{0}_{\ell} = \sum_{j \in [d]} \beta_j X_j \;\; .$$

But this is equivalent to $\boldsymbol{y} - \boldsymbol{x} \notin \mathrm{span}(Y) \cup \mathrm{span}(X)$ (see Lemma 1) which holds if and only if the predicate is true, as needed.

All the polynomials in the key and the ciphertext which have not been used for reconstructability can be eliminated. Figure 2 describes the resulting encoding after this simplification.

A.2 Arguing security

Our Theorem 2 guarantees that the encoding from Figure 3 is secure. Note that removing polynomials cannot change security (only spoil reconstructability), so

PES for predicate $P((\boldsymbol{x}, X), (\boldsymbol{y}, Y)) = 1 \Leftrightarrow (\boldsymbol{x} + \mathrm{span}(X)) \cap (\boldsymbol{y} + \mathrm{span}(Y)) = \emptyset$.

Where $\mathcal{X} := \mathbb{Z}_N^d \times \mathbb{Z}_N^{d \times \ell}$ and $\mathcal{Y} := \mathbb{Z}_N^d \times \mathbb{Z}_N^{d \times \ell'}$ for integers N, d, ℓ, ℓ'.

- $\mathsf{Param}(\mathsf{par}) \to d+5$ and let $\boldsymbol{b} = (b_0, \boldsymbol{b}', t, u, v, w)$ with $\boldsymbol{b}' = (b_1, \ldots, b_d)$.
- $\mathsf{EncKey}(N, (\boldsymbol{y}, Y)) :=$
  $\{\, r_1(b_0 + w),\ r_1(b_0 + t),\ r_2 t,\ r_2 v,\ r_1 u,\ r_2 u + \hat{r}_1,\ r_1 v + \hat{r}_1,$
  $(Y_j \hat{\boldsymbol{r}}' + r_2 b_j + \hat{r}_1 y_j)_{j \in [d]},\ (r_1 b_j)_{j \in [d]} \,\}$ (and also $\alpha + r_1 b_0$).
- $\mathsf{EncCt}(N, (\boldsymbol{x}, X)) :=$
  $\{\, s_1 w,\ s_1 t,\ s_0 t - \hat{s}_1,\ s_1 v - \hat{s}_1,\ s_1 u - \hat{s}_2,\ s_0 u,\ s_0 v$
  $(X_j \hat{\boldsymbol{s}}' - s_1 b_j + \hat{s}_2 x_j)_{j \in [d]},\ (s_0 b_j)_{j \in [d]} \,\}$.

Here, $\boldsymbol{r} := (r_1, r_2)$, $\hat{\boldsymbol{r}} := (\hat{r}_1, \hat{\boldsymbol{r}}')$ with $\hat{\boldsymbol{r}}' := (\hat{r}_2, \ldots, \hat{r}_{\ell'+1})$, and $\boldsymbol{s} := (s_0, s_1)$, $\hat{\boldsymbol{s}} := (\hat{s}_1, \hat{s}_2, \hat{\boldsymbol{s}}')$ with $\hat{\boldsymbol{s}}' := (\hat{s}_3, \ldots, \hat{s}_{\ell+2})$.

Fig. 3. PES for negated doubly spatial encryption.

the simpler scheme presented in the main body (Figure 2) must also be secure. Nevertheless, we provide an independent proof of its security, for the sake of completeness.

Proof (Security of the encoding from Figure 2). Assume the predicate is false, i.e., the affine spaces $\boldsymbol{x} + \mathrm{span}(X)$ and $\boldsymbol{y} + \mathrm{span}(Y)$ intersect. Let $\boldsymbol{z} \in \mathbb{Z}_N^d$ be a vector in their intersection and let $\boldsymbol{z}_{\boldsymbol{x}} \in \mathbb{Z}_N^\ell$ and $\boldsymbol{z}_{\boldsymbol{y}} \in \mathbb{Z}_N^{\ell'}$ be such that:

$$\boldsymbol{x} + X\boldsymbol{z}_{\boldsymbol{x}} = \boldsymbol{z} = \boldsymbol{y} + Y\boldsymbol{z}_{\boldsymbol{y}} \ .$$

Observe that all the polynomials in $\mathsf{EncKey}(N, (\boldsymbol{y}, Y))$ and $\mathsf{EncCt}(N, (\boldsymbol{x}, X))$ (see Figure 2) evaluate to zero on the following substitution:

$$(\boldsymbol{b}, \hat{\boldsymbol{r}}', \hat{\boldsymbol{s}}') \leftarrow (\boldsymbol{z}, \boldsymbol{z}_y, \boldsymbol{z}_x) \qquad r_1, s_1, \hat{r}_1, \hat{s}_2, u, t, \alpha \leftarrow 1 \qquad b_0, s_0, r_2, \hat{s}_1, v \leftarrow -1 \ ,$$

but polynomial αs_0 evaluates to -1 ($\neq 0$). As explained in Example 1, this is an evidence of the security of the encoding.

B Additional definitions

B.1 Security of attribute-based encryption

An ABE scheme is *adaptively secure* if there exists a negligible ϵ such that for all PPT adversaries $\mathcal{A}$, and all sufficiently large $\lambda \in \mathbb{N}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}}(\lambda) < \epsilon(\lambda)$, where:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}}(\lambda) := \Pr\left[\begin{array}{r} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}) \\ x^\star \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk}, \cdot)}(\mathsf{mpk}) \\ (\mathsf{ct}_{x^\star}, \tau) \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^\star) \\ b \xleftarrow{\$} \{0,1\};\ \tau_0 := \tau;\ \tau_1 \xleftarrow{\$} \mathcal{K} \\ b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk}, \cdot)}(\mathsf{ct}_{x^\star}, \tau_b) \end{array} : \ b' = b \right] - \frac{1}{2}$$

where the advantage is defined to be zero if some of the queries y made by $\mathcal{A}$ to the KeyGen oracle violates the condition $P(x^\star, y) = 0$.

B.2 Attribute-based encryption from pair encodings

In order to explain how to build attribute-based encryption from pair encodings, we need to introduce the notion of dual system groups (DSG) [CW13, CW14, AC17], since the compilers from pair encodings into ABE [Att16, AC16] rely on DSG in a black-box way.

Dual System Groups

A dual system group is a tuple of six efficiently computable algorithms:

- $\mathsf{SampP}(1^\lambda, 1^n)$: on input the security parameter and an integer n, outputs public parameters pp and secret parameters sp such that:
  - The public parameters, pp, include a triple of abelian groups $(\mathsf{G}, \mathsf{H}, \mathsf{G_t})$ (that are $\mathbb{Z}_p$-modules for some λ-bits prime p), a non-degenerate bilinear map $e : \mathsf{G} \times \mathsf{H} \to \mathsf{G_t}$, an homomorphism μ (defined over H) and additional parameters required by SampP and SampH.
  - Given pp, it is possible to uniformly sample to H.
  - The secret parameters, sp, include a distinguished element $h^* \in \mathsf{H}$ (different from the unit) and additional parameters required by $\widehat{\mathsf{SampG}}$ and $\widehat{\mathsf{SampH}}$.
- $\mathsf{SampG}(\mathsf{pp})$ and $\widehat{\mathsf{SampG}}(\mathsf{pp}, \mathsf{sp})$ output an element from G^{n+1}.
- $\mathsf{SampH}(\mathsf{pp})$ and $\widehat{\mathsf{SampH}}(\mathsf{pp}, \mathsf{sp})$ output an element from H^{n+1}.
- SampGT is a function defined from $\mathrm{Im}(\mu)$ to $\mathsf{G_t}$.

Additional conditions are required for correctness and security:

projective: For all public parameters, pp, every $h \in \mathsf{H}$ and all coin tosses σ, it holds $\mathsf{SampGT}(\mu(h); \sigma) = e(g_0, h)$, where $(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathsf{pp}; r)$.

associative: Let $(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathsf{pp})$, $(h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\mathsf{pp})$, it holds $e(g_0, h_i) = e(g_i, h_0)$ for every $i \in [n]$.

H-subgroup: $\mathsf{SampH}(\mathsf{pp})$ is the uniform distribution over a subgroup of H^{n+1}.

orthogonality: $h^* \in \mathrm{Kernel}(\mu)$.

non-degeneracy: For every $(h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\mathsf{pp})$, $h^* \in \langle h_0 \rangle$. Furthermore, for every $(\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_n) \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp}, \mathsf{sp})$, $(\alpha \xleftarrow{\$} \mathbb{Z}_p; \mathsf{return}\, e(\hat{g}_0, h^*)^\alpha)$ is the uniform distribution over $\mathsf{G_t}$.

left-subgroup indistinguishability: $(\mathsf{pp}, \boldsymbol{g}) \approx_c (\mathsf{pp}, \boldsymbol{g}\cdot\hat{\boldsymbol{g}})$.

right-subgroup indistinguishability: $(\mathsf{pp}, h^*, \boldsymbol{g}\cdot\hat{\boldsymbol{g}}, \boldsymbol{h}) \approx_c (\mathsf{pp}, h^*, \boldsymbol{g}\cdot\hat{\boldsymbol{g}}, \boldsymbol{h}\cdot\hat{\boldsymbol{h}})$.

parameter-hiding: $(\mathsf{pp}, h^*, \hat{\boldsymbol{g}}, \hat{\boldsymbol{h}}) \equiv (\mathsf{pp}, h^*, \hat{\boldsymbol{g}}\cdot\hat{\boldsymbol{g}}', \hat{\boldsymbol{h}}\cdot\hat{\boldsymbol{h}}')$.

Where, $\approx_c$ denotes a distinguishing probability upper-bounded by a negligible function on λ and, for any $n \in \mathbb{N}$, the above elements are sampled as:

$$(\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n)$$

$$\begin{array}{lll} \boldsymbol{g} \leftarrow \mathsf{SampG}(\mathsf{pp}) & \hat{\boldsymbol{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp}, \mathsf{sp}) & \hat{\boldsymbol{g}}' \coloneqq (1_{\mathsf{G}}, \hat{\boldsymbol{g}}_0^{z_1}, \ldots, \hat{\boldsymbol{g}}_0^{z_n}) \\ \boldsymbol{h} \leftarrow \mathsf{SampG}(\mathsf{pp}) & \hat{\boldsymbol{h}} \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp}, \mathsf{sp}) & \hat{\boldsymbol{h}}' \coloneqq (1_{\mathsf{H}}, \hat{\boldsymbol{h}}_0^{z_1}, \ldots, \hat{\boldsymbol{h}}_0^{z_n}) \end{array}$$

for $z_1, \ldots, z_n \xleftarrow{\$} \mathbb{Z}_p$.

Remark. Observe that we have presented the version of dual system groups defined in [CGW15]. Other works consider slightly different conditions (e.g., the non-degeneracy of [AC16]). However, the widely used instantiation of DSG from k-lin given in [CGW15] also satisfies the properties of those variations.

ABE from pair encodings

Given a pair encoding scheme $\{\mathsf{Param}, \mathsf{EncKey}, \mathsf{EncCt}, \mathsf{Pair}\}$ (see Definition 2) for a predicate family $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ indexed by $\kappa = (N, \mathsf{par})$ (let $\lambda = |N|$), an attribute-based encryption scheme can be constructed as follows:

- $\mathsf{Setup}(1^\lambda, \mathcal{X}_\kappa, \mathcal{Y}_\kappa)$: let $n \leftarrow \mathsf{Param}(\mathsf{par})$ and run the DSG generation algorithm $\mathsf{SampP}(1^\lambda, 1^n)$ to obtain pp and sp. Let $\mathsf{msk} \xleftarrow{\$} \mathsf{H}$ and $\mathsf{mpk} \coloneqq (\mathsf{pp}, \mu(\mathsf{msk}))$. Output $(\mathsf{mpk}, \mathsf{msk})$.
- $\mathsf{Enc}(\mathsf{mpk}, x)$: run $\mathsf{EncCt}(N, x)$ to obtain polynomials $\boldsymbol{c}_x(\boldsymbol{s}, \hat{\boldsymbol{s}}, \boldsymbol{b})$. For every $\ell \in [w_3]$, let the ℓ-th polynomial in $\boldsymbol{c}_x$ be

  $$\sum_{i\in[w_2]} \gamma_i^{(\ell)} \hat{s}_i + \sum_{i\in[0,w_1-1]} \sum_{j\in[n]} \gamma_{\{i,j\}}^{(\ell)} s_i b_j$$

  for some coefficients $\gamma_i^{(\ell)}$ and $\gamma_{\{i,j\}}^{(\ell)}$ in $\mathbb{Z}_p$. Now, run SampG to produce

  $$\begin{array}{ll} (\hat{g}_{\{i,0\}}, \hat{g}_{\{i,1\}}, \ldots, \hat{g}_{\{i,n\}}) \leftarrow \mathsf{SampG}(\mathsf{pp}) & \text{for } i \in [w_2] \\ (g_{\{i,0\}}, g_{\{i,1\}}, \ldots, g_{\{i,n\}}) \leftarrow \mathsf{SampG}(\mathsf{pp}) & \text{for } i \in [0, w_1-1] \\ (g_{\{0,0\}}, g_{\{0,1\}}, \ldots, g_{\{0,n\}}) \leftarrow \mathsf{SampG}(\mathsf{pp}; \sigma) & \end{array}$$

  Observe that we have made explicit the coin tosses, σ, used in the last sampling. Setup $\mathsf{ct}_x \coloneqq \left(\mathsf{ct}_0, \mathsf{ct}_1, \ldots, \mathsf{ct}_{w_1-1}, \widetilde{\mathsf{ct}}_1, \ldots, \widetilde{\mathsf{ct}}_{w_3}\right)$ and define the

symmetric encryption key as $\tau \coloneqq \mathsf{SampGT}(\mu(\mathsf{msk}); \sigma)$, where $\mathsf{ct}_i \coloneqq g_{\{i,0\}}$ for every $i \in [0, w_1 - 1]$; and for every $\ell \in [w_3]$, $\widetilde{\mathsf{ct}}_\ell$ is computed as

$$\widetilde{\mathsf{ct}}_\ell \coloneqq \prod_{i \in [w_2]} \hat{g}_{\{i,0\}}^{\gamma_i^{(\ell)}} \cdot \prod_{i \in [0, w_1 - 1]} \prod_{j \in [n]} g_{\{i,j\}}^{\gamma_{\{i,j\}}^{(\ell)}} \ .$$

Output (ct_x, τ).

- $\mathsf{KeyGen}(\mathsf{msk}, y)$: run $\mathsf{EncKey}(N, y)$ to obtain polynomials $\boldsymbol{k}_y(\boldsymbol{r}, \hat{\boldsymbol{r}}, \boldsymbol{b})$. For every $\ell \in [m_3]$, let the ℓ-th polynomial in $\boldsymbol{k}_y$ be

$$\phi^{(\ell)} \alpha + \sum_{i \in [m_2]} \phi_i^{(\ell)} \hat{r}_i + \sum_{i \in [m_1]} \sum_{j \in [n]} \phi_{\{i,j\}}^{(\ell)} r_i b_j$$

for some coefficients $\phi^{(\ell)}$, $\phi_i^{(\ell)}$ and $\phi_{\{i,j\}}^{(\ell)}$ in $\mathbb{Z}_p$. Now, run SampH to produce

$$\begin{aligned} (\hat{h}_{\{i,0\}}, \hat{h}_{\{i,1\}}, \ldots, \hat{h}_{\{i,n\}}) &\leftarrow \mathsf{SampH}(\mathsf{pp}) \qquad \text{for } i \in [m_2] \\ (h_{\{i,0\}}, h_{\{i,1\}}, \ldots, h_{\{i,n\}}) &\leftarrow \mathsf{SampH}(\mathsf{pp}) \qquad \text{for } i \in [m_1] \end{aligned}$$

Define the secret key as $\mathsf{sk}_y \coloneqq \left(\mathsf{sk}_1, \ldots, \mathsf{sk}_{m_1}, \widetilde{\mathsf{sk}}_1, \ldots, \widetilde{\mathsf{sk}}_{m_3}\right)$, where $\mathsf{sk}_i \coloneqq h_{\{i,0\}}$ for every $i \in [m_1]$; and for every $\ell \in [m_3]$, $\widetilde{\mathsf{sk}}_\ell$ is computed as

$$\widetilde{\mathsf{sk}}_\ell \coloneqq \mathsf{msk}^{\phi^{(\ell)}} \cdot \prod_{i \in [m_2]} \hat{h}_{\{i,0\}}^{\phi_i^{(\ell)}} \cdot \prod_{i \in [m_1]} \prod_{j \in [n]} h_{\{i,j\}}^{\phi_{\{i,j\}}^{(\ell)}} \ .$$

Output sk_y.

- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{ct}_x, x)$: run $\mathsf{Pair}(N, x, y)$ to obtain matrices E, E' (note that y is assumed to be extractable from sk_y, whereas x is explicitly included as an input to Dec). Define:

$$\tau \coloneqq \prod_{i \in [w_1]} \prod_{\ell \in [m_3]} e(\mathsf{ct}_{i-1}, \widetilde{\mathsf{sk}}_\ell)^{E_{i,\ell}} \cdot \prod_{\ell \in [w_3]} \prod_{i \in [m_1]} e(\widetilde{\mathsf{ct}}_\ell, \mathsf{sk}_i)^{E'_{\ell,i}}$$

Output the symmetric encryption key τ.