

Generic Transformations of Predicate Encodings: Constructions and Applications

Miguel Ambrona^{1,2(✉)}, Gilles Barthe¹, and Benedikt Schmidt³

¹ IMDEA Software Institute, Madrid, Spain

`{miguel.ambrona,gilles.barthe}@imdea.org`

² Universidad Politécnica de Madrid, Madrid, Spain

³ Google, Mountain View, USA

`beschmidt@google.com`

Abstract. Predicate encodings (Wee, TCC 2014; Chen, Gay, Wee, EUROCRYPT 2015), are symmetric primitives that can be used for building predicate encryption schemes. We give an algebraic characterization of the notion of privacy from predicate encodings, and explore several of its consequences. Specifically, we propose more efficient predicate encodings for boolean formulae and arithmetic span programs, and generic optimizations of predicate encodings. We define new constructions to build boolean combination of predicate encodings. We formalize the relationship between predicate encodings and pair encodings (Attrapadung, EUROCRYPT 2014), another primitive that can be transformed generically into predicate encryption schemes, and compare our constructions for boolean combinations of pair encodings with existing similar constructions from pair encodings. Finally, we demonstrate that our results carry to tag-based encodings (Kim, Susilo, Guo, and Au, SCN 2016).

1 Introduction

Predicate Encryption (PE) [13, 25] is a form of public-key encryption that supports fine-grained access control for encrypted data. In predicate encryption, everyone can create ciphertexts while keys can only be created by the master key owner. Predicate encryption schemes use predicates to model (potentially complex) access control policies, and attributes are attached to both ciphertexts and secret keys. A predicate encryption scheme for a predicate P guarantees that decryption of a ciphertext ct_x with a secret key sk_y is allowed if and only if the attribute x associated to the ciphertext ct and the attribute y associated to the secret key sk verify the predicate P , i.e. $P(x, y) = 1$. Predicate encryption schemes exist for several useful predicates, such as Zero Inner Product Encryption (ZIPE), where attributes are vectors \mathbf{x} and \mathbf{y} and the predicate $P(\mathbf{x}, \mathbf{y})$ is defined as $\mathbf{x}^\top \mathbf{y} = 0$. Predicate encryption subsumes several previously defined notions of public-key encryption. For example, Identity-Based Encryption (IBE) [34] can be obtained by defining $P(x, y)$ as $x = y$ and Attribute-Based Encryption (ABE) [33] can be obtained similarly. More concretely, for Key-Policy

ABE (KP-ABE), the attribute x is a boolean vector, the attribute y is a boolean function, and the predicate $P(x, y)$ is defined as $y(x) = 1$. For Ciphertext-Policy ABE (CP-ABE), the roles of the attributes x and y are swapped.

Modular Approaches for PE. In 2014, two independent works by Wee [37] and Attrapadung [6] proposed generic and unifying frameworks for obtaining efficient fully secure PE schemes for a large class of predicates. Both works use the dual system methodology introduced by Lewko and Waters [27, 36] and define a compiler that takes as input a relatively simple symmetric primitive and produces a fully secure PE construction. Wee introduced so-called *predicate encodings*, an information-theoretic primitive inspired from linear secret sharing. Attrapadung introduced so-called *pair encodings* and provided computational and information-theoretic security notions. These approaches greatly simplify the construction and analysis of predicate encryption schemes and share several advantages. First, they provide a good trade-off between *expressivity* and *performance*, while the security relies on standard and well studied assumptions. Second, they unify existing constructions into a single framework, i.e., previous PE constructions can be seen as instantiations of these new compilers with certain encodings. Third, building PE schemes by analyzing and building these simpler encodings is much easier than building PE schemes directly. Compared to full security for PE, the encodings must verify much weaker security requirements. The power of pair and predicate encodings is evidenced by the discovery of new constructions and efficiency improvements over existing ones. However, both approaches were designed over *composite order* bilinear groups. In Chen et al. [15] and Attrapadung [7] respectively extended the predicate encoding and pair encoding compiler to the *prime order* setting. Next, Agrawal and Chase [1] improved on Attrapadung’s work by relaxing the security requirement on *pair encodings* and thus, capturing new constructions. In addition, their work also brings both generic approaches closer together, because like in [15], the new compiler relies (in a black-box way) on Dual System Groups (DSG) [16, 17]. Additionally Kim et al. [22] recently introduced a new generic framework for modular design of predicate encryption that improves on the performance of existing compilers. Their core primitive, *tag-based encodings*, is very similar to *predicate encodings*.

1.1 Our Contributions

We pursue the study of predicate encodings and establish several general results and new constructions that broaden their scope and improve their efficiency. We also compare predicate encodings to pair and tag-based encodings.

Predicate Encodings. We show that the information-theoretic definition of α -privacy used in [15, 37] is equivalent to an algebraic statement (furthermore independent of α) about the existence of solutions for a linear system of equations. Leveraging this result, we prove a representation theorem for predicate

encodings: every triple of encoding functions implicitly defines a unique predicate for which it is a valid predicate encoding. Conversely, every predicate P that admits a predicate encoding is logically equivalent to the implicit predicate induced by its encoding functions. Moreover, our algebraic definition of privacy simplifies all subsequent results in the paper.

First, we define a generic optimization of predicate encodings that often leads to efficiency improvements and reduce the number of required group elements in keys and ciphertexts. We prove the soundness of the transformations and validate their benefits experimentally on examples from [15, 37]; we successfully apply these simplifications to reduce the size of keys and ciphertexts by up to 50% and to reduce the number of group operations needed in some of the existing encodings.

Second, we define generic methods for combining predicate encodings. We provide encoding transformations for the *disjunction*, *conjunction* and *negation* of predicates, and for the *dual* predicate.

Tag-Based Encodings. We show that our results on predicate encodings generalize to tag-based encodings. In particular, we give a purely algebraic characterization of the hiding property of tag-based encodings. Moreover, we demonstrate that the hiding property can be strengthened without any loss of generality, by requiring equality rather than statistical closeness of distributions.

Comparison of Encodings. We compare the expressivity of the three core primitives (*predicate encodings*, *pair encodings* and *tag-based encodings*) corresponding to the three different modular frameworks. We provide an embedding that produces an information-theoretical pair encoding from every predicate encoding. Then, we use this encoding to compare our constructions to build boolean combination of predicate encodings with similar constructions for pair encodings that were introduced by [6].

In addition, we provide a transformation¹ from tag-based encodings into predicate encodings.

New Constructions. We develop several new constructions of predicate encodings and predicate encryption:

- **Combining predicates.** We show how to combine our results to build *Dual-Policy Attribute-Based Encryption (DP-ABE)* [9, 10] in the frameworks of predicate encodings and tag-based encodings (Sect. 6.1). Additionally, we consider the idea of combining arbitrary encodings with a *broadcast encryption* encoding to achieve direct revocation of keys. The former encoding takes care of revocation, while the latter encodes the desired access structure.
- **Improved predicate encodings.** We provide new instances of predicate encodings that improve on best known predicate encodings proposed in [15] and have additional properties (Sect. 6.2).

¹ This transformation has side conditions, thus it is not universal, but all existing tag-based encodings (except one) satisfy these side conditions.

- **Extra features.** Finally, we show how to construct a weakly attribute-hiding predicate encoding for boolean formulas and how to enhance any predicate encoding with support for delegation (Sect. 6.3).

Implementation and Evaluation. We implement a general library for predicate encryption with support for the predicate encoding and pair encoding frameworks. Our library uses the Relic-Toolkit [5] for pairings with a 256-bits Barreto-Naehrig Curve [11]. We use our library for validating our constructions; experimental results are presented in the relevant sections. All the experiments were executed on a 8-core machine with 2.40 GHz Intel Core i7-3630QM CPU and 8 GB of RAM. Our scalability experiments show that predicate encodings can be used for real applications. The code is publicly available and open source².

1.2 Prior Work

Predicate encodings have been introduced in [37] and we use a refined version that is defined in [15] as our starting point. Both variants use an information-theoretic definition of the hiding while we show that there is an equivalent algebraic definition. Another related work is [20], initiating a systematic study of the communication complexity of the so-called conditional secret disclosure primitive, which is closely related to predicate encodings.

Other works also optimize existing predicate encryption schemes, for example many works focus on going from composite order constructions to the more efficient prime order ones [7, 15, 26]. In [15] they also propose performance improvements on dual system groups. We believe our optimizations via predicate encodings complement other possible enhancements of predicate encryption.

Boolean combinations of predicates have also been considered in the setting of pair encodings. Attrapadung [9, 10] proposes generic transformations for conjunction and for the dual predicate, but neither for negation nor disjunction. We propose new transformations for conjunction and dual in the framework of predicate encodings and we also deal with negation and disjunction.

The main advantage of DP-ABE is the possibility of considering policies over *objective* attributes (associated to data) and policies over *subjective* attributes (associated to user credentials) at the same time. DP-ABE has been considered by Attrapadung in the pair encoding framework [9, 10]. To the best of our knowledge, we are the first to provide DP-ABE in the predicate encoding and tag-based encoding frameworks.

Revocation is a desirable property for PE and ABE schemes that has also been considered by many works in the literature. Revocation allows to invalidate a user's secret key in such a way that it becomes useless, even if its associated attribute satisfies the policy associated to the ciphertext. Some attempts [32] propose *indirect* revocation that requires that the master secret owner periodically updates secret keys for non-revoked users. Other attempts achieve *direct* revocation [8, 23, 30, 31], but either rely on strong assumptions or provide only

² Source code at <https://github.com/miguel-ambrona/abe-relic>.

selectively security. Our construction not only allows to achieve revocation in a *fully secure* framework, but it allows to add revocation to arbitrary predicate encodings.

Policy hiding is another property of PE, and ABE in particular, that has been broadly studied. In this context, policies associated to ciphertexts are not attached to them and therefore, unauthorized users will only learn the fact that their key does not satisfy the policy, but nothing else. Policy Hiding has been considered in several works [13, 25]. The security of our construction improves on earlier works, thanks to the compiler from [15]. Our observation extends the expressivity of attribute-hiding predicate encryption for ZIPE proposed in [15] to support policy-hiding for boolean formulas.

In [15], the authors introduce the notion of *spatial encryption* predicate encodings. We generalize this notion and introduce a transformation that makes delegation possible for every predicate encoding.

Several works evaluate the suitability of ABE for different applications. For example, ABE has been used and benchmarked to enforce privacy of Electronic Medical Records (EMR) [3], in a system where healthcare organizations export EMRs to external storage locations. Other examples are Sieve [35] or Streamforce [18], systems that provide enforced access control for user data and stream data in untrusted clouds. In contrast to these works, we are the first to evaluate predicate encryption and ABE based on modern modular approaches such as the predicate encoding and pair encoding frameworks. The resulting schemes also satisfy a stronger security notion (full vs. selective security) compared to the previously mentioned evaluations. We focus on synthetic case studies, while other works analyze more realistic settings and integration of ABE into bigger systems. Combining our methods with these more practical case studies is a very interesting direction for future work.

1.3 Comparison with Agrawal and Chase (EUROCRYPT 2017)

Concurrently and independently, Agrawal and Chase [2] introduce a new security notion, which they call *symbolic property*, for pair encodings. They adapt previous generic frameworks [1, 7] 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 assumption that is implied by some *q-type* assumptions proposed in [6, 29]. Moreover, they introduce the notion of *trivially broken* pair encoding and show that any not trivially broken pair encoding must satisfy their symbolic property. Their definitions of symbolic property and trivially broken for pair encodings are closely related to our algebraic characterization of privacy of predicate encodings. However, the two results are incomparable: although pair encodings are more general than predicate encodings (see Sect. 5.1 for a more detailed comparison), their results rely on *q-type* assumption, whereas our results build on previous frameworks that rely on weaker assumptions (Matrix-DH or k-LIN).

2 Background

In this section, we first introduce some mathematical notation and then define *predicate encodings*, *tag-based encodings* and *pair encodings* the three primitives used in the three different modular frameworks for predicate encryption.

2.1 Notation

For finite sets S , we use $x \xleftarrow{\$} S$ to denote that x is uniformly sampled from S . We define $[n]$ as the range $\{1, \dots, n\}$ for an arbitrary $n \in \mathbb{N}$. For a predicate $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$, we use $(x, y) \in P$ as a shorthand for $P(x, y) = 1$. We use the same conventions for matrix-representations of linear maps on finite-dimensional spaces. We define vectors $\mathbf{v} \in \mathbb{F}^n$ as column matrices and denote the *transpose* of a matrix A by A^\top . We use $\text{diag}(\mathbf{v})$ to denote the diagonal matrix with main diagonal \mathbf{v} . 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 $\mathbf{0}_{m,n}$. Let S be a set of indices and A be a matrix. A_S denotes the matrix formed from the set of columns of A with indices in S . We define the *colspan* of a matrix $M \in \mathbb{F}^{m \times n}$ as the set of all possible linear combinations columns of M . That is $\text{colspan}(M) = \{M\mathbf{v} : \mathbf{v} \in \mathbb{F}^n\} \subseteq \mathbb{F}^m$. We analogously define the *rowspan* of a matrix. We consider prime order bilinear groups $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_t)$ and use g_1, g_2, g_t to denote their respective generators. The map e satisfies $e(g_1^a, g_2^b) = g_t^{ab}$ for every $a, b \in \mathbb{N}$. A bilinear group is said to be symmetric if $\mathbb{G}_1 = \mathbb{G}_2$, otherwise it is called asymmetric. We abuse of notation and write $g^{\mathbf{v}}$ to denote $(g^{v_1}, \dots, g^{v_n})$ for a group element g and a vector $\mathbf{v} \in \mathbb{Z}_p^n$.

2.2 Predicate Encodings

Predicate encodings are information-theoretic primitives that can be used for building predicate encryption schemes [37]. We adopt the definition from [15], but prefer to use matrix notation.

Definition 1 (Predicate encoding). *Let $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ be a binary predicate over finite sets \mathcal{X} and \mathcal{Y} . Given a prime $p \in \mathbb{N}$, and $s, r, w \in \mathbb{N}$, a (s, r, w) -predicate encoding for P consists of five deterministic algorithms ($\text{sE}, \text{rE}, \text{kE}, \text{sD}, \text{rD}$): the sender encoding algorithm sE maps $x \in \mathcal{X}$ into a matrix $\text{sE}_x \in \mathbb{Z}_p^{s \times w}$, the receiver encoding algorithm rE maps $y \in \mathcal{Y}$ into a matrix $\text{rE}_y \in \mathbb{Z}_p^{r \times w}$, the key encoding algorithm kE maps $y \in \mathcal{Y}$ into a vector $\text{kE}_y \in \mathbb{Z}_p^r$, while the sender and receiver decoding algorithms, respectively sD and rD , map a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ into vectors $\text{sD}_{x,y} \in \mathbb{Z}_p^s$ and $\text{rD}_{x,y} \in \mathbb{Z}_p^r$ respectively. We require that the following properties are satisfied:*

reconstructability: for all $(x, y) \in P$, $\text{sD}_{x,y}^\top \text{sE}_x = \text{rD}_{x,y}^\top \text{rE}_y$ and $\text{rD}_{x,y}^\top \text{kE}_y = 1$;
 α -privacy: for all $(x, y) \notin P, \alpha \in \mathbb{Z}_p$,

$$\mathbf{w} \xleftarrow{\$} \mathbb{Z}_p^w; \text{ return } (\text{sE}_x \mathbf{w}, \text{rE}_y \mathbf{w} + \alpha \cdot \text{kE}_y) \equiv \mathbf{w} \xleftarrow{\$} \mathbb{Z}_p^w; \text{ return } (\text{sE}_x \mathbf{w}, \text{rE}_y \mathbf{w})$$

where \equiv denotes equality of distributions.

Reconstructability allows to recover α from $(x, y, \mathbf{sE}_x \mathbf{w}, \mathbf{rE}_y \mathbf{w} + \alpha \cdot \mathbf{kE}_y)$ if $(x, y) \in \mathcal{P}$. *Privacy* ensures that α is perfectly hidden for such tuples if $(x, y) \notin \mathcal{P}$.

Example 1 (IBE predicate encoding). Let $\mathcal{X} = \mathcal{Y} = \mathbb{Z}_p$ and let $s = r = 1, w = 2$. We define the encoding functions as follows:

$$\begin{aligned} \mathbf{sE}_x &= \begin{pmatrix} x & 1 \end{pmatrix} & \mathbf{sD}_{x,y} &= \begin{pmatrix} 1 \end{pmatrix} \\ \mathbf{rE}_y &= \begin{pmatrix} y & 1 \end{pmatrix} & \mathbf{rD}_{x,y} &= \begin{pmatrix} 1 \end{pmatrix} \\ \mathbf{kE}_y &= \begin{pmatrix} 1 \end{pmatrix} \end{aligned}$$

The above is a predicate encoding for identity-based encryption, i.e., for the predicate $\mathcal{P}(x, y) = 1$ iff $x = y$. Note that $\begin{pmatrix} x & 1 \end{pmatrix} = \begin{pmatrix} y & 1 \end{pmatrix}$ when $x = y$, so reconstructability is satisfied. On the other hand, α -privacy follows from the fact that if $x \neq y$, $x \cdot w_1 + w_2$ and $y \cdot w_1 + w_2$ are pair-wise independent. ■

Predicate Encryption from Predicate Encodings. We try to provide some intuition on how predicate encodings are compiled to predicate encryption schemes by the compiler from [15]. We consider a simplified compiler (see explanations below). The master keys, ciphertexts and secret keys have the following form:

$$\begin{aligned} \mathbf{msk} &= g_2^\alpha & \mathbf{ct}_x &= (g_1^s, g_1^{s \cdot \mathbf{sE}_x \mathbf{w}}, e(g_1, g_2)^{\alpha s} \cdot m) \\ \mathbf{mpk} &= (g_1, g_1^w, g_2, g_2^w, e(g_1, g_2)^\alpha) & \mathbf{sk}_y &= (g_2^r, g_2^{\alpha \cdot \mathbf{kE}_y + r \cdot \mathbf{rE}_y \mathbf{w}}) \end{aligned}$$

The encrypted message $m \in \mathbb{G}_t$ is blinded by a random factor $e(g_1, g_2)^{\alpha s}$. The so-called *reconstruction* property of predicate encodings ensures that this blinding factor can be recovered for a pair $(\mathbf{ct}_x, \mathbf{sk}_y)$ if $\mathcal{P}(x, y) = 1$. More concretely, for all pairs (x, y) such that $\mathcal{P}(x, y) = 1$, because multiplying by matrices $\mathbf{sD}_{x,y}, \mathbf{rD}_{x,y}$ is a linear operation, it is possible operate in the exponent and compute

$$g_1^{s \cdot \mathbf{sD}_{x,y}^\top \mathbf{sE}_x \mathbf{w}} \quad \text{and} \quad g_2^{\mathbf{rD}_{x,y}^\top (\alpha \cdot \mathbf{kE}_y + r \cdot \mathbf{rE}_y \mathbf{w})},$$

obtaining $g_1^{s\beta}$ and $g_2^{\alpha+r\beta}$ for $\beta = \mathbf{sD}_{x,y}^\top \mathbf{sE}_x \mathbf{w} = \mathbf{rD}_{x,y}^\top \mathbf{rE}_y \mathbf{w}$ (note that knowing the value of β is not necessary). Now, it is simple to recover $e(g_1, g_2)^{\alpha s}$ from $e(g_1^s, g_2^{\alpha+r\beta})$ and $e(g_1^{s\beta}, g_2^r)$. Security is ensured by the α -privacy property of the encoding together with decisional assumptions about dual system groups. Intuitively, the α -privacy property states that given certain values derived from the output of the encoding functions for random input, α remains information-theoretic hidden.

Note that the following is a simplification of their compiler, where we avoid DSG for simplicity. The real scheme produced by their compiler would have twice as many group elements (under SXDH) or three times as many (under DLIN).

2.3 Tag-Based Encodings

Tag-based encodings is a new primitive defined in a very recent work [22] that defines a new generic framework (using prime order groups) for modular design of predicate encryption.

Definition 2 (Tag-based encoding). Let $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ be a binary predicate over finite sets \mathcal{X} and \mathcal{Y} . Given a prime $p \in \mathbb{N}$, and $c, k, h \in \mathbb{N}$, a (c, k, h) -tag-based encoding for P consists of two deterministic algorithms (cE, kE) : the ciphertext encoding algorithm cE maps $x \in \mathcal{X}$ into a matrix $\text{cE}_x \in \mathbb{Z}_p^{c \times h}$ and the key encoding algorithm kE maps $y \in \mathcal{Y}$ into a matrix $\text{kE}_y \in \mathbb{Z}_p^{k \times h}$. We require that the following properties are satisfied:

reconstructability: for all $(x, y) \in P$, there exists an efficient algorithm that on input (x, y) computes vectors $\mathbf{m}_c \in \mathbb{Z}_p^c$, $\mathbf{m}_k \in \mathbb{Z}_p^k$ such that

$$\mathbf{m}_c^\top \text{cE}_x = \mathbf{m}_k^\top \text{kE}_y \neq \mathbf{0}_h^\top$$

h-hiding: for all $(x, y) \notin P$,

$$\mathbf{h} \xleftarrow{\$} \mathbb{Z}_p^h; \text{ return } (\text{cE}_x \mathbf{h}, \text{kE}_y \mathbf{h}) \approx_s \mathbf{h}, \mathbf{h}' \xleftarrow{\$} \mathbb{Z}_p^h; \text{ return } (\text{cE}_x \mathbf{h}, \text{kE}_y \mathbf{h}')$$

where \approx_s denotes negligible statistical distance between distributions.

The compiler proposed in [22] uses similar ideas to the one for predicate encodings. However, it does not rely on dual system groups and can be instantiated with symmetric bilinear maps. The message is blinded and ciphertexts and keys contain a set of group elements that are enough to recover the blinding factor only when the predicate is true. This compiler has the advantage that some elements of ciphertexts and keys are \mathbb{Z}_p values and not group elements, which reduces the storage size.

2.4 Pair Encodings

Attrapadung [6, 7] proposes an independent modular framework for predicate encryption, based on a primitive called *pair encoding*. For our purposes, it suffices to consider a more restrictive, information-theoretic, notion of pair encodings.

Definition 3 (Information-theoretic pair encoding). Let $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ be a binary predicate over finite sets \mathcal{X} and \mathcal{Y} . Given a prime $p \in \mathbb{N}$, and $c, k, l, m, n \in \mathbb{N}$, let $\mathbf{h} = (h_1, \dots, h_n)$, $\mathbf{s} = (s_0, s_1, \dots, s_l)$ and $\mathbf{r} = (\alpha, r_1, \dots, r_m)$ be sets of variables. An information-theoretic (c, k, n) -pair encoding scheme for P consists of three deterministic algorithms $(\text{Enc1}, \text{Enc2}, \text{Pair})$: the ciphertext encoding algorithm Enc1 maps a value $x \in \mathcal{X}$ into a list of polynomials $\mathbf{c}_x \in \mathbb{Z}_p[\mathbf{s}, \mathbf{h}]^c$, the key encoding algorithm Enc2 maps a value $y \in \mathcal{Y}$ into a list of polynomials $\mathbf{k}_y \in \mathbb{Z}_p[\mathbf{r}, \mathbf{h}]^k$ and the decoding algorithm Pair maps a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ into a matrix $E_{x,y} \in \mathbb{Z}_p^{c \times k}$. We require that the following properties are satisfied:

polynomial constraints:

- For every $x \in \mathcal{X}$ and every $f \in \text{Enc1}(x)$, $f = f(\mathbf{s}, \mathbf{h})$ only contains monomials of the form s_i or $s_i h_j$, $i \in [0, l]$, $j \in [n]$.

- For every $y \in \mathcal{Y}$ and every $f \in \text{Enc2}(y)$, $f = f(\mathbf{r}, \mathbf{h})$ only contains monomials of the form α , r_i or $r_i h_j$, $i \in [m]$, $j \in [n]$.

reconstructability: for all $(x, y) \in \mathcal{P}$ and all $\mathbf{c}_x \leftarrow \text{Enc1}(x)$, $\mathbf{k}_y \leftarrow \text{Enc2}(y)$, $E_{x,y} \leftarrow \text{Pair}(x, y)$, the following polynomial equality holds $\mathbf{c}_x^\top E_{x,y} \mathbf{k}_y = \alpha s_0$.

perfect security: for all $(x, y) \notin \mathcal{P}$ and all $\mathbf{c}_x \leftarrow \text{Enc1}(x)$, $\mathbf{k}_y \leftarrow \text{Enc2}(y)$,

$$\begin{aligned} \mathbf{h} &\stackrel{\$}{\leftarrow} \mathbb{Z}_p^n; \mathbf{r} \stackrel{\$}{\leftarrow} (\mathbb{Z}_p^*)^m; \mathbf{s} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^{l+1}; & \text{return } (\mathbf{c}_x(\mathbf{s}, \mathbf{h}), \mathbf{k}_y(0, \mathbf{r}, \mathbf{h})) &\equiv \\ \mathbf{h} &\stackrel{\$}{\leftarrow} \mathbb{Z}_p^n; \mathbf{r} \stackrel{\$}{\leftarrow} (\mathbb{Z}_p^*)^m; \mathbf{s} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^{l+1}; \alpha \stackrel{\$}{\leftarrow} \mathbb{Z}_p; & \text{return } (\mathbf{c}_x(\mathbf{s}, \mathbf{h}), \mathbf{k}_y(\alpha, \mathbf{r}, \mathbf{h})) \end{aligned}$$

where \equiv denotes equality of distributions.

The compiler from pair encodings follows similar ideas to the other compilers. The message is blinded by a random factor and ciphertexts and keys contain all the information necessary to recover this blinded factor, only when the predicate holds. The compiler from pair encodings requires to compute a polynomial number of pairings during decryption, unlike the compilers for predicate encodings and tag-based encodings that need³ 6 and 8 pairings respectively.

3 Predicate Encodings: Properties and Consequences

In this section, we present a purely algebraic (and independent of α) characterization of the α -privacy property. It simplifies both the analysis and the construction of predicate encodings. In particular, we use our characterization to define and prove a new optimization of predicate encodings, i.e., a transformation that makes the encoding functions smaller while preserving the predicate. Additionally, we unify the reconstructability and privacy properties and show that they are mutually exclusive and complementary, i.e., for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$, one and only one of the two conditions holds. This unified treatment facilitates the construction and study of predicate encodings.

3.1 Algebraic Properties of Predicate Encodings

The following theorem captures two essential properties of predicate encodings: first, privacy admits a purely algebraic characterization (furthermore independent of α) given in terms of existence of solutions of a linear system of equations. Second, reconstructability and privacy, when viewed as properties of a single pair (x, y) , negate each other; i.e. a pair (x, y) always satisfies exactly one of the two properties.

Theorem 1 (Algebraic characterization of privacy). *Let $p \in \mathbb{N}$ be a prime, let $s, r, w \in \mathbb{N}$ and let $S \in \mathbb{Z}_p^{s \times w}$, $R \in \mathbb{Z}_p^{r \times w}$, $\mathbf{k} \in \mathbb{Z}_p^r$. The following are equivalent:*

³ Decryption in the framework of predicate encodings needs 4 pairings under SXDH assumption or 6 under DLIN, in the framework of tag-based encodings decryption requires 8 pairings and the assumption is DLIN.

- **α -privacy** For every $\alpha \in \mathbb{Z}_p$,

$$\mathbf{w} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^w; \text{ return } (S\mathbf{w}, R\mathbf{w} + \alpha \cdot \mathbf{k}) \quad \equiv \quad \mathbf{w} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^w; \text{ return } (S\mathbf{w}, R\mathbf{w})$$

- **(algebraic) privacy** There exists $\mathbf{w} \in \mathbb{Z}_p^w$ such that $S\mathbf{w} = \mathbf{0}_s$ and $R\mathbf{w} = \mathbf{k}$
- **non-reconstructability** For every $\mathbf{s} \in \mathbb{Z}_p^s$ and $\mathbf{r} \in \mathbb{Z}_p^r$, either $\mathbf{s}^\top S \neq \mathbf{r}^\top R$ or $\mathbf{r}^\top \mathbf{k} \neq 1$.

Proof. We first prove that α -privacy is equivalent to algebraic privacy. Note that the fact that $\forall \alpha \in \mathbb{Z}_p$,

$$\mathbf{w} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^w; \text{ return } (S\mathbf{w}, R\mathbf{w} + \alpha \cdot \mathbf{k}) \quad \equiv \quad \mathbf{w} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^w; \text{ return } (S\mathbf{w}, R\mathbf{w})$$

is equivalent to the existence of a bijection ρ_α such that for all $\mathbf{w} \in \mathbb{Z}_p^w$, $S\mathbf{w} = S \cdot \rho_\alpha(\mathbf{w}) \wedge R\mathbf{w} + \alpha \cdot \mathbf{k} = R \cdot \rho_\alpha(\mathbf{w})$. By linearity, it can be rewritten as

$$S(\rho_\alpha(\mathbf{w}) - \mathbf{w}) = \mathbf{0}_s \quad \wedge \quad \alpha \cdot \mathbf{k} = R(\rho_\alpha(\mathbf{w}) - \mathbf{w})$$

Now, the existence of such a bijection is equivalent to the existence of a solution for the following (parametric in α) linear system on \mathbf{w} : $S\mathbf{w} = \mathbf{0}_s \wedge R\mathbf{w} = \alpha \cdot \mathbf{k}$. To see this, note that if ρ_α is such a bijection, $\rho_\alpha(\mathbf{w}_0) - \mathbf{w}_0$ is a solution of the system for every $\mathbf{w}_0 \in \mathbb{Z}_p^w$. On the other hand, if \mathbf{w}^* is a solution of the system, the bijection $\rho_\alpha(\mathbf{w}) = \mathbf{w} + \mathbf{w}^*$ satisfies the required identities. To conclude, note that the above system has a solution iff the following (independent of α) does:

$$S\mathbf{w} = \mathbf{0}_s \quad \wedge \quad R\mathbf{w} = \mathbf{k}$$

Next, we prove the equivalence between algebraic privacy and non-reconstructability. We use the following helping lemma from [12, Claim 2]: for every field \mathbb{F} , let $A \in \mathbb{F}^{m \times n}$ and $\mathbf{b} \in \mathbb{F}^n$ be matrices with coefficients in \mathbb{F} , the following two statements are equivalent:

- for every $\mathbf{a} \in \mathbb{F}^m$, $\mathbf{b}^\top \neq \mathbf{a}^\top A$;
- there exists $\mathbf{z} \in \mathbb{F}^n$ such that $\mathbf{z}^\top \mathbf{b} = 1$ and $A\mathbf{z} = \mathbf{0}_m$.

Assume that algebraic privacy does not hold, i.e., for every $\mathbf{w} \in \mathbb{Z}_p^w$, either $S\mathbf{w} \neq \mathbf{0}_s$ or $R\mathbf{w} \neq \mathbf{k}$. Equivalently, for every $\mathbf{w} \in \mathbb{Z}_p^w$

$$\begin{pmatrix} \mathbf{0}_s \\ \mathbf{k} \end{pmatrix} \neq \begin{pmatrix} -S \\ R \end{pmatrix} \mathbf{w}$$

which is equivalent (by our helping lemma) to the existence of $(\mathbf{z}_1, \mathbf{z}_2) \in \mathbb{Z}_p^s \times \mathbb{Z}_p^r$ such that

$$(\mathbf{z}_1^\top \ \mathbf{z}_2^\top) \begin{pmatrix} \mathbf{0}_s \\ \mathbf{k} \end{pmatrix} = 1 \quad \wedge \quad (\mathbf{z}_1^\top \ \mathbf{z}_2^\top) \begin{pmatrix} -S \\ R \end{pmatrix} = \mathbf{0}_w^\top$$

That is, there exists $\mathbf{z}_1 \in \mathbb{Z}_p^s$, $\mathbf{z}_2 \in \mathbb{Z}_p^r$ such that $\mathbf{z}_1^\top S = \mathbf{z}_2^\top R \wedge \mathbf{z}_2^\top \mathbf{k} = 1$, which is exactly reconstructability. The proof follows from the fact all the steps are equivalences. \square

Our next result is a representation theorem. It is based on the notion of partial encoding; informally, a partial encoding consists of the first three algorithms of a predicate encoding; it is not attached to any specific predicate, nor is required to satisfy any property.

Definition 4 (Partial encoding). *Let \mathcal{X} and \mathcal{Y} be finite sets. Let $p \in \mathbb{N}$ be a prime and $s, r, w \in \mathbb{N}$. A (s, r, w) -partial encoding is given by three deterministic algorithms (sE, rE, kE) : sE maps $x \in \mathcal{X}$ into a matrix $sE_x \in \mathbb{Z}_p^{s \times w}$, and rE, kE map $y \in \mathcal{Y}$ into a matrix $rE_y \in \mathbb{Z}_p^{r \times w}$ and a vector $kE_y \in \mathbb{Z}_p^r$ respectively.*

The representation theorem shows that there exists an embedding from partial encodings to predicate encodings, and that every predicate encoding lies the image of the embedding.

Theorem 2 (Representation theorem). *Let \mathcal{X} and \mathcal{Y} be finite sets. Let $p \in \mathbb{N}$ be a prime and $s, r, w \in \mathbb{N}$. Every (s, r, w) -partial encoding (sE, rE, kE) for \mathcal{X} and \mathcal{Y} induces a predicate encoding (sE, rE, kE, sD, rD) for the following predicate (henceforth coined implicit predicate):*

$$\text{Pred}(x, y) \triangleq \forall \mathbf{w} \in \mathbb{Z}_p^w, \quad sE_x \mathbf{w} \neq \mathbf{0}_s \vee rE_y \mathbf{w} \neq kE_y$$

Moreover, if (sE, rE, kE, sD, rD) is a predicate encoding for P , then for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $P(x, y) \Leftrightarrow \text{Pred}(x, y)$.

Example 2 (Implicit predicate of IBE predicate encoding). If we consider the following partial encoding functions corresponding to the encoding presented in Example 1:

$$sE_x = \begin{pmatrix} x & 1 \end{pmatrix} \qquad rE_y = \begin{pmatrix} y & 1 \end{pmatrix} \qquad kE_y = \begin{pmatrix} 1 \end{pmatrix}$$

our Theorem 2 guarantees that it is a valid predicate encoding for the implicit predicate:

$$\text{Pred}(x, y) = 1 \text{ iff } \forall (w_1, w_2) \in \mathbb{Z}_p^2, \quad x \cdot w_1 + w_2 \neq 0 \vee y \cdot w_1 + w_2 \neq 1$$

A simple analysis shows that the above predicate is equivalent to $x = y$. ■

A consequence of Theorem 2 is that a predicate P over \mathcal{X} and \mathcal{Y} can be instantiated by a (s, r, w) -predicate encoding iff there exist \mathcal{X} -indexed and \mathcal{Y} -indexed matrices $S_x \in \mathbb{Z}_p^{s \times w}$ and $R_y \in \mathbb{Z}_p^{r \times w}$ and \mathcal{Y} -indexed vectors $\mathbf{k}_y \in \mathbb{Z}_p^r$ such that:

$$P(x, y) = 1 \text{ iff } \begin{pmatrix} \mathbf{0}_s \\ \mathbf{k}_y \end{pmatrix} \notin_{\text{span}}^{\text{col}} \begin{pmatrix} S_x \\ R_y \end{pmatrix}$$

That is helpful to analyze the expressivity of predicate encodings of certain size.

Example 3. Let \mathcal{X} and \mathcal{Y} be finite sets, let $n \in \mathbb{N}$, we will characterize all the predicates that can be achieved from a $(1, 1, n)$ -partial encoding, say (sE, rE, kE) .

Note that for every pair (x, y) , \mathbf{sE}_x and \mathbf{rE}_y are vectors of length n , while \mathbf{kE}_y is a single element. Say,

$$\mathbf{sE}_x = (f_1(x), \dots, f_n(x)) \quad \mathbf{rE}_y = (g_1(y), \dots, g_n(y)) \quad \mathbf{kE}_y = h(y)$$

for certain functions $f_i : \mathcal{X} \rightarrow \mathbb{Z}_p$, $g_i, h : \mathcal{Y} \rightarrow \mathbb{Z}_p$ for every $i \in [n]$. Theorem 2 guarantees that the above is a valid predicate encoding for the predicate

$$P(x, y) = 1 \text{ iff } h(y) \neq 0 \wedge \left(\exists \beta \in \mathbb{Z}_p : \bigwedge_{i \in [n]} f_i(x) = \beta g_i(y) \right)$$

It can be shown that the predicate $P((x_1, x_2), y) = 1$ iff $(x_1 = y) \vee (x_2 = y)$ cannot be captured by $(1, 1, n)$ -predicate encodings, while on the contrary, the predicate $P((x_1, x_2), y) = 1$ iff $(x_1 = y) \wedge (x_2 = y)$ could be instantiated. ■

3.2 Optimizing Predicate Encodings

In this section, we show that the efficiency of predicate encodings can be improved by pre- and post-processing. Specifically, we show that an (s, r, w) -encoding $(\mathbf{sE}, \mathbf{rE}, \mathbf{kE}, \mathbf{sD}, \mathbf{rD})$ for a predicate P can be transformed into a (s', r', w') -encoding $(\mathbf{sE}', \mathbf{rE}', \mathbf{kE}', \mathbf{sD}', \mathbf{rD}')$ for the same predicate, by applying a linear transformation to the matrices induced by $\mathbf{sE}, \mathbf{rE}, \mathbf{kE}$.

More precisely, if we define $\mathbf{sE}'_x = A\mathbf{sE}_x$, $\mathbf{rE}'_y = B\mathbf{rE}_y$ and $\mathbf{kE}'_y = B\mathbf{kE}_y$ for two matrices A and B , the privacy of the encoding will be preserved, but reconstructability may be destroyed. On the contrary, when we consider the partial encoding $\mathbf{sE}'_x = \mathbf{sE}_x C$, $\mathbf{rE}'_y = \mathbf{rE}_y C$ and $\mathbf{kE}'_y = \mathbf{kE}_y$ for a matrix C , reconstructability is automatically guaranteed, but privacy could not hold (for the same predicate). Intuitively, this occurs because *reconstructability* depends on the *rowspan* of the matrices $\mathbf{sE}_x, \mathbf{rE}_y$, while *privacy* depends on their *colspan*. Our following theorem imposes conditions on these matrices A, B and C so that the resulting predicate encoding is equivalent to the original one.

Theorem 3. *Let \mathcal{X} and \mathcal{Y} be finite sets. Let $p \in \mathbb{N}$ be a prime, $s, r, w, s', r', w' \in \mathbb{N}$, and let $(\mathbf{sE}, \mathbf{rE}, \mathbf{kE}, \mathbf{sD}, \mathbf{rD})$ be a (s, r, w) -predicate encoding for $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$. Let A be a function that maps every element $x \in \mathcal{X}$ into a matrix $A_x \in \mathbb{Z}_p^{s \times s}$, B be a function that maps $y \in \mathcal{Y}$ into a matrix $B_y \in \mathbb{Z}_p^{r' \times r}$ and let $C \in \mathbb{Z}_p^{w \times w'}$ be a matrix. There exists a (s', r', w') -partial encoding $(\mathbf{sE}', \mathbf{rE}', \mathbf{kE}', \mathbf{sD}', \mathbf{rD}')$ for P , where*

$$\mathbf{sE}'_x = A_x \mathbf{sE}_x C \quad \mathbf{rE}'_y = B_y \mathbf{rE}_y C \quad \mathbf{kE}'_y = B_y \mathbf{kE}_y$$

provided the following conditions hold:

- For all $(x, y) \in P$, $\mathbf{sD}_{x,y}^\top \in \text{row}_{\text{span}} \langle A_x \rangle$ and $\mathbf{rD}_{x,y}^\top \in \text{row}_{\text{span}} \langle B_y \rangle$;
- For all $(x, y) \notin P$, there exists $\mathbf{w} \in \text{col}_{\text{span}} \langle C \rangle$ s.t. $\mathbf{sE}_x \mathbf{w} = \mathbf{0}_s$ and $\mathbf{rE}_y \mathbf{w} = \mathbf{kE}_y$.

This transformation is useful to make predicate encodings simpler and more efficient in different manners. For instance, it can be used to make the matrices corresponding to encoding and decoding functions become sparser. That is, if we consider \mathbf{A} and \mathbf{B} as functions that apply matrix *Gaussian elimination*⁴ to the matrices associated to \mathbf{sE} and \mathbf{rE}, \mathbf{kE} , many entries from these matrices will be zero. Hence, fewer group operations will be performed during encryption and key generation, improving the performance. Moreover, the transformation can be used to reduce the size of \mathbf{mpk} , \mathbf{ct}_x and \mathbf{sk}_y . If $w' < w$, the number of elements in \mathbf{mpk} will decrease. This will also improve the performance of encryption and key generation (both depend directly on \mathbf{mpk}). Additionally, if $s' < s$ or $r' < r$, the number of elements in \mathbf{ct}_x and \mathbf{sk}_y will also decrease respectively.

Note that a simplification from the right (multiplying by \mathbf{C}) changes the structure of the encoding and may open the possibility of left-simplifications that were not available before and vice versa. Example 4 illustrates this idea. We optimize a predicate encoding that corresponds to the result of applying our negation transformation (from next section, Theorem 6) to the predicate encoding from Example 1.

Example 4. Let $\mathcal{X} = \mathcal{Y} = \mathbb{Z}_p$ and consider the $(2, 3, 4)$ -predicate encoding $(\mathbf{sE}, \mathbf{rE}, \mathbf{kE}, \mathbf{sD}, \mathbf{rD})$ for $P(x, y) = 1$ iff $x \neq y$, defined as

$$\mathbf{sE}_x = \begin{pmatrix} x-1 & 0 & 0 \\ 1 & 0 & -1 & 0 \end{pmatrix} \quad \mathbf{rE}_y = \begin{pmatrix} 0 & 1 & 0 & y \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{kE}_y = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\mathbf{sD}_{x,y}^\top = \begin{pmatrix} \frac{-1}{x-y} & \frac{x}{x-y} \end{pmatrix} \quad \mathbf{rD}_{x,y}^\top = \begin{pmatrix} \frac{1}{x-y} & \frac{-x}{x-y} & 1 \end{pmatrix}$$

Note that for every pair $(x, y) \notin P$, i.e. $x = y$, the single solution of the system $\mathbf{sE}_x \mathbf{w} = \mathbf{0}_2 \wedge \mathbf{rE}_y \mathbf{w} = \mathbf{kE}_y$ is $\mathbf{w}^\top = (-1 \ -y \ -1 \ 1)$, thus the matrix

$$\mathbf{C} = \begin{pmatrix} -1 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}^\top$$

satisfies the conditions of Theorem 3. Therefore, the $(2, 3, 2)$ -partial encoding $(\mathbf{sE}', \mathbf{rE}', \mathbf{kE}')$, where

$$\mathbf{sE}'_x = \mathbf{sE}_x \mathbf{C} = \begin{pmatrix} -x & -1 \\ 0 & 0 \end{pmatrix} \quad \mathbf{rE}'_y = \mathbf{rE}_y \mathbf{C} = \begin{pmatrix} y & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \mathbf{kE}'_y = \mathbf{kE}_y = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

induces a predicate encoding for the same predicate. The previous simplification, opens the possibility of applying again the theorem, with matrices \mathbf{A}_x and \mathbf{B}_y , obtaining a $(1, 2, 2)$ -predicate encoding for $P(x, y) = 1$ iff $x \neq y$. Concretely,

⁴ Note that if matrices \mathbf{A}_x , \mathbf{B}_y or \mathbf{C} are invertible, they always satisfy their respective requirements.

$$\begin{aligned}
A_x &= \begin{pmatrix} -1 & 0 \end{pmatrix} & sE''_x &= \begin{pmatrix} x & 1 \end{pmatrix} & rE''_y &= \begin{pmatrix} y & 1 \\ 1 & 0 \end{pmatrix} & rE''_y &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
B_y &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} & sD''_{x,y} &= \begin{pmatrix} 1 \\ x-y \end{pmatrix} & rD''_{x,y} &= \begin{pmatrix} 1 \\ x-y & 1 \end{pmatrix}
\end{aligned}$$

The above simplifications can be successfully applied to actual predicate encodings proposed in [15]. In Sect. 6.2 we propose improved predicate encodings for *monotonic boolean formulas* and *arithmetic span programs*. ■

3.3 Combining Predicates

Using the new characterization of predicate encodings from the previous section, we define transformations to combine predicate encodings into new predicate encodings for more complex predicates. In particular, we define predicate encoding transformations for disjunction, conjunction, negation and the dual predicate. These combinations are useful to create new schemes that inherit different properties from the more basic building blocks. In Sect. 6, we propose several constructions that rely on these transformations.

Disjunction. We present a method to build a predicate encoding for the disjunction of P_1 and P_2 from predicate encodings for P_1 and P_2 . Observe that the predicate encryption scheme obtained from the resulting predicate encoding is more efficient than the predicate encryption scheme obtained by compiling the predicate encodings of P_1 and P_2 separately, and then applying a generic transformation that builds predicate encryption schemes for a disjunction from predicate encryption schemes of its disjuncts.

Theorem 4 (Disjunction of predicate encodings). *For every (s_1, r_1, w_1) -predicate encoding $(sE^1, rE^1, kE^1, sD^1, rD^1)$ for $P_1 : \mathcal{X}_1 \times \mathcal{Y}_1 \rightarrow \{0, 1\}$ and every (s_2, r_2, w_2) -predicate encoding $(sE^2, rE^2, kE^2, sD^2, rD^2)$ for $P_2 : \mathcal{X}_2 \times \mathcal{Y}_2 \rightarrow \{0, 1\}$, there exists a $(s_1 + s_2, r_1 + r_2, w_1 + w_2)$ -predicate encoding (sE, rE, kE, sD, rD) for the predicate $P : (\mathcal{X}_1, \mathcal{X}_2) \times (\mathcal{Y}_1, \mathcal{Y}_2) \rightarrow \{0, 1\}$ such that:*

$$P((x_1, x_2), (y_1, y_2)) \Leftrightarrow P_1(x_1, y_1) \vee P_2(x_2, y_2)$$

Concretely,

$$sE_{(x_1, x_2)} = \begin{pmatrix} sE^1_{x_1} & \mathbf{0}_{s_1, w_2} \\ \mathbf{0}_{s_2, w_1} & sE^2_{x_2} \end{pmatrix} \quad rE_{(y_1, y_2)} = \begin{pmatrix} rE^1_{y_1} & \mathbf{0}_{r_1, w_2} \\ \mathbf{0}_{r_2, w_1} & rE^2_{y_2} \end{pmatrix} \quad kE_{(y_1, y_2)} = \begin{pmatrix} kE^1_{y_1} \\ kE^2_{y_2} \end{pmatrix}$$

$$\begin{aligned}
sD^{\top}_{(x_1, x_2), (y_1, y_2)} &= \text{if } P_1(x_1, y_1) \text{ then } (sD^1_{x_1, y_1} \mathbf{0}_{s_2}^{\top}) \text{ else } (\mathbf{0}_{s_1}^{\top} sD^2_{x_2, y_2}^{\top}) \\
rD^{\top}_{(x_1, x_2), (y_1, y_2)} &= \text{if } P_1(x_1, y_1) \text{ then } (rD^1_{x_1, y_1} \mathbf{0}_{r_2}^{\top}) \text{ else } (\mathbf{0}_{r_1}^{\top} rD^2_{x_2, y_2}^{\top})
\end{aligned}$$

Note that it is possible to obtain sharing between attributes, e.g., if $\mathcal{X}_1 = \mathcal{X}_2$ and the sender uses only the subset $\{(x, x) \mid x \in \mathcal{X}_1\}$ of $\mathcal{X}_1 \times \mathcal{X}_2$, the predicate becomes $P(x, (y_1, y_2)) = 1$ iff $P_1(x, y_1) \vee P_2(x, y_2)$.

Conjunction. In contrast to disjunction, the naive solution that just concatenates secret keys fails. Given keys for attribute pairs (y_1, y_2) and (y'_1, y'_2) , it would be possible to recombine the components and obtain a key for (y_1, y'_2) leading to collusion attacks. Our predicate encoding transformation deals with this problem by “tying” the two components together with additional randomness.

Theorem 5 (Conjunction of predicate encodings). *For every (s_1, r_1, w_1) -predicate encoding $(sE^1, rE^1, kE^1, sD^1, rD^1)$ for $P_1 : \mathcal{X}_1 \times \mathcal{Y}_1 \rightarrow \{0, 1\}$ and every (s_2, r_2, w_2) -predicate encoding $(sE^2, rE^2, kE^2, sD^2, rD^2)$ for $P_2 : \mathcal{X}_2 \times \mathcal{Y}_2 \rightarrow \{0, 1\}$, there exists a $(s_1 + s_2, r_1 + r_2, w_1 + w_2 + 1)$ -predicate encoding (sE, rE, kE, sD, rD) for the predicate $P : (\mathcal{X}_1, \mathcal{X}_2) \times (\mathcal{Y}_1, \mathcal{Y}_2) \rightarrow \{0, 1\}$ such that:*

$$P((x_1, x_2), (y_1, y_2)) \Leftrightarrow P_1(x_1, y_1) \wedge P_2(x_2, y_2)$$

Concretely,

$$\begin{aligned} sE_{(x_1, x_2)} &= \begin{pmatrix} sE_{x_1}^1 & \mathbf{0}_{s_1, w_2} & \mathbf{0}_{s_1} \\ \mathbf{0}_{s_2, w_1} & sE_{x_2}^2 & \mathbf{0}_{s_2} \end{pmatrix} & sD_{(x_1, x_2), (y_1, y_2)} &= \frac{1}{2} \begin{pmatrix} sD_{x_1, y_1}^1 \\ sD_{x_2, y_2}^2 \end{pmatrix} \\ rE_{(y_1, y_2)} &= \begin{pmatrix} rE_{y_1}^1 & \mathbf{0}_{r_1, w_2} & kE_{y_1}^1 \\ \mathbf{0}_{r_2, w_1} & rE_{y_2}^2 & -kE_{y_2}^2 \end{pmatrix} & rD_{(x_1, x_2), (y_1, y_2)} &= \frac{1}{2} \begin{pmatrix} rD_{x_1, y_1}^1 \\ rD_{x_2, y_2}^2 \end{pmatrix} \\ kE_{(y_1, y_2)} &= \begin{pmatrix} kE_{y_1}^1 \\ kE_{y_2}^2 \end{pmatrix} \end{aligned}$$

Note that it is possible to combine Theorems 4 and 5 to create a predicate encoding for $P_1 \bowtie P_2$, where the placeholder $\bowtie \in \{\vee, \wedge\}$ can be part of keys or ciphertexts.

Negation. To obtain a functionally complete set of boolean predicate encoding transformers, we now define a transformation for negation. Our transformation unifies negated predicates like Non-zero Inner Product Encryption (NIPE) and Zero Inner Product Encryption (ZIPE). In Sect. 6.2 we use this transformation to build optimized predicate encodings. The technique works for predicate encodings where the negation transformation yields a predicate encoding that can be further simplified (using our method from Sect. 3.2).

Theorem 6 (Negation of predicate encodings). *For every (s, r, w) -predicate encoding (sE, rE, kE, sD, rD) for $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ there exists a $(w, w + 1, s + w + r)$ -predicate encoding $(sE', rE', kE', sD', rD')$ for the predicate $P' : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ such that $P'(x, y) \Leftrightarrow \neg P(x, y)$. Concretely,*

$$\begin{aligned} sE'_x &= \begin{pmatrix} sE_x^\top & -I_w & \mathbf{0}_{w, r} \end{pmatrix} & rE'_y &= \begin{pmatrix} \mathbf{0}_{w, s} & I_w & rE_y^\top \\ \mathbf{0}_s^\top & \mathbf{0}_w^\top & kE_y^\top \end{pmatrix} & kE'_y &= \begin{pmatrix} \mathbf{0}_w \\ 1 \end{pmatrix} \\ sD'_{x, y} &= w_{x, y} & rD'_{x, y} &= \begin{pmatrix} -w_{x, y} \\ 1 \end{pmatrix} \end{aligned}$$

where for a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $P(x, y) = 0$, $\mathbf{w}_{x,y}$ is defined as the witness for algebraic privacy, i.e., a vector such that

$$\mathbf{sE}_x \mathbf{w}_{x,y} = \mathbf{0}_s \quad \wedge \quad \mathbf{rE}_y \mathbf{w}_{x,y} = \mathbf{kE}_y$$

Note that such a vector always exists when $P(x, y) = 0$. Moreover, \mathbf{sD} and \mathbf{rD} do not need to be defined when $P'(x, y)$ is not 1, that is, when $P(x, y)$ is not 0.

A similar construction has been considered in a posterior work [4] to this work. Specifically, they show how to transform a *conditional disclosure of secrets* (CDS) for f into a CDS for \bar{f} (the complement of f).

Dual. In the literature, the notions of KP-ABE and CP-ABE are considered separately. In fact, many works are only valid for one of the two versions of Attribute Based Encryption. Our transformation unifies the notion of KP-ABE and CP-ABE in the framework of predicate encodings. In this context they should not be considered separately, because our transformation provides a Ciphertext-Policy predicate encoding from any Key-Policy predicate encoding and vice versa.

Theorem 7 (Dual predicate encoding). *For every (s, r, w) -predicate encoding $(\mathbf{sE}, \mathbf{rE}, \mathbf{kE}, \mathbf{sD}, \mathbf{rD})$ for $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ there exists a $(r, s + 1, w + 1)$ -predicate encoding $(\mathbf{sE}', \mathbf{rE}', \mathbf{kE}', \mathbf{sD}', \mathbf{rD}')$ for the predicate $P' : \mathcal{Y} \times \mathcal{X} \rightarrow \{0, 1\}$ such that $P'(y, x) \Leftrightarrow P(x, y)$. Concretely,*

$$\begin{aligned} \mathbf{sE}'_y &= (\mathbf{rE}_y \quad \mathbf{kE}_y) & \mathbf{rE}'_x &= \begin{pmatrix} \mathbf{sE}_x & \mathbf{0}_s \\ \mathbf{0}_w^\top & 1 \end{pmatrix} & \mathbf{kE}'_x &= \begin{pmatrix} \mathbf{0}_s \\ 1 \end{pmatrix} \\ \mathbf{sD}'_{y,x} &= \mathbf{rD}_{x,y} & \mathbf{rD}'_{y,x} &= \begin{pmatrix} \mathbf{sD}_{x,y} \\ 1 \end{pmatrix} \end{aligned}$$

4 Tag-Based Encodings

We show that our techniques for *predicate encodings* can be extended to the framework of *tag-based encodings*. In particular, we show a similar result to our Theorem 1, which establishes that \mathbf{h} -hiding and reconstructability are mutually exclusive and complementary.

Theorem 8. *Let $p \in \mathbb{N}$ be a prime, let $k, c, h \in \mathbb{N}$ and let $C \in \mathbb{Z}_p^{c \times h}$, $K \in \mathbb{Z}_p^{k \times h}$. The following are equivalent:*

- **\mathbf{h} -hiding:** $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_p^h$; $\text{return } (C\mathbf{h}, K\mathbf{h}) \equiv \mathbf{h}, \mathbf{h}' \xleftarrow{\$} \mathbb{Z}_p^h$; $\text{return } (C\mathbf{h}, K\mathbf{h}')$
- **non-reconstructability** For every $\mathbf{m}_c \in \mathbb{Z}_p^c$ and very $\mathbf{m}_k \in \mathbb{Z}_p^k$, either $\mathbf{m}_c^\top C \neq \mathbf{m}_k^\top K$ or $\mathbf{m}_c^\top C = \mathbf{0}_h^\top$.

where \equiv denotes equality of distributions.

A consequence of Theorem 8 is that every valid tag-based encoding is perfectly hiding, or equivalently, there cannot exist a tag-based encoding where the two distributions from \mathbf{h} -hiding are negligibly close but not identical.

Thanks to the above theorem, it is possible to define *disjunction* and *conjunction* transformations for tag-based encodings along the lines of predicate encodings. We were not able to design a negation transformation for tag-based encodings and leave it for future work. On the other hand, the dual transformation is straightforward in this framework, as mentioned in [22], because the encoding primitives are completely symmetric.

Expressivity of Tag-Based Encodings vs Predicate Encodings. We propose a transformation that produces valid predicate encodings from valid tag-based encodings for the same predicate.

Theorem 9. *Given a $(c, 1, h)$ -tag-based encoding (cE, kE) for $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$, the $(c, 1, h)$ -partial predicate encoding (sE', rE', kE') defined as $sE'_x = cE_x$, $rE'_y = kE_y$, $kE'_y = (1)$, induces a predicate encoding for P .*

Note that because of the symmetry of tag-based encodings, Theorem 9 can be also applied to $(1, k, h)$ -tag-based encodings. All the tag-based encodings proposed in [22] (except one) have either $c = 1$ or $k = 1$, so the above theorem can be applied to them.

5 Pair Encodings

In this section we provide an embedding that transforms every predicate encoding into an information-theoretic pair encoding. Consequently, we can see predicate encodings as a subclass of pair encodings. This opens the possibility of reusing the conjunction and dual transformation proposed by Attrapadung [9, 10] for pair encodings, to create combinations of predicate encodings via our embedding. We show that this alternative method is fundamentally different from our direct conjunction and dual transformations on predicate encodings, where our combinations produce more efficient encodings.

5.1 Embedding Predicate Encodings into Pair Encodings

In this section we provide an embedding that produces a valid *information-theoretic pair encoding* from every valid predicate encoding (see Definitions 1 and 3 for predicate encodings and pair encodings respectively).

Definition 5 (Embedding to Pair Encodings). *Given a (s, r, w) -predicate encoding $pe = (sE, rE, kE, sD, rD)$, we define the embedding $\text{Emb}(pe) = (\text{Enc1}_{pe}, \text{Enc2}_{pe}, \text{Pair}_{pe})$ as follows:*

- $\text{Enc1}_{pe}(x) = (c_0, \mathbf{c})$, where $c_0(s_0, \mathbf{h}) = s_0$, $\mathbf{c}(s_0, \mathbf{h}) = s_0 \cdot sE_x \mathbf{h}$
- $\text{Enc2}_{pe}(y) = (k_0, \mathbf{k})$, where $k_0(\alpha, r_1, \mathbf{h}) = r_1$, $\mathbf{k}(\alpha, r_1, \mathbf{h}) = \alpha \cdot kE_y + r_1 \cdot rE_y \mathbf{h}$

$$\bullet \text{ Pair}_{pe}(x, y) = \begin{pmatrix} 0 & rD_{x,y}^\top \\ -sD_{x,y} & 0_{s,r} \end{pmatrix}$$

All variables $\mathbf{s} = (s_0)$ and $\mathbf{r} = (r_1)$ appear in the clear in the **Enc1** and **Enc2** polynomials respectively. This simplifies the pair encoding's information-theoretical security notion into one equivalent to the privacy of the predicate encoding (see proof of Theorem 10).

Theorem 10 (Correctness of the embedding). *If $pe = (sE, rE, kE, sD, rD)$ is a valid (s, r, w) -predicate encoding for P , then $\text{Emb}(pe)$ is a valid information theoretic $(s + 1, r + 1, w)$ -pair encoding for P .*

Our embedding shows that every predicate encoding can be transformed into a perfectly secure pair encoding. In fact, after applying the compiler from [1] to the embedding of a predicate encoding, we get the same predicate encryption scheme that the one provided by the compiler from [15].

We conclude that *predicate encodings* can be transformed into a very special class of *pair encodings*: encodings that allow decryption with 2 pairings and have only one element of randomness in both, ciphertexts and secret keys (what makes them very efficient).

5.2 Comparison Between Encoding Transformations

Attrapadung proposed generic transformations of pair encodings [9, 10]. In particular, he proposed the conjunction and dual transformations. In this section we compare these transformations with the ones proposed in this work. For this, we compare the conjunction of two pair encodings, (embedded from predicate encodings) with the embedding of the conjunction of a (s_1, r_1, w_1) -predicate encoding $pe^1 = (sD^1, rE^1, kE^1, sD^1, rD^1)$ and a (s_2, r_2, w_2) -predicate encoding $pe^2 = (sD^2, rE^2, kE^2, sD^2, rD^2)$, i.e.,

$$\text{Emb}(pe^1 \wedge_{pred} pe^2) \text{ vs } \text{Emb}(pe^1) \wedge_{pair} \text{Emb}(pe^2)$$

where \wedge_{pred} and \wedge_{pair} are the conjunction of predicate encodings and pair encodings respectively. Note that \wedge_{pred} corresponds to the transformation from our Theorem 5. On the other hand, for \wedge_{pair} we use the conjunction proposed in [10].

$$\text{Emb}(pe^1 \wedge_{pred} pe^2) = \begin{cases} \text{Enc1}((x_1, x_2)) = (c_0, \mathbf{c}_1, \mathbf{c}_2) \\ \text{Enc2}((y_1, y_2)) = (k_0, \mathbf{k}_1, \mathbf{k}_2) \\ \text{Pair}((x_1, x_2), (y_1, y_2)) = E_{(x_1, x_2), (y_1, y_2)} \end{cases}$$

where $\mathbf{h} = (h_0, \mathbf{h}_1, \mathbf{h}_2)$ and

$$\begin{aligned} c_0(s_0, \mathbf{h}) &= s_0 & k_0(\alpha, r_1, \mathbf{h}) &= r_1 \\ \mathbf{c}_1(s_0, \mathbf{h}) &= s_0 \cdot sE_{x_1}^1 \mathbf{h}_1 & \mathbf{k}_1(\alpha, r_1, \mathbf{h}) &= (\alpha + h_0) \cdot kE_{y_1}^1 + r_1 \cdot rE_{y_1}^1 \mathbf{h}_1 \\ \mathbf{c}_2(s_0, \mathbf{h}) &= s_0 \cdot sE_{x_2}^2 \mathbf{h}_2 & \mathbf{k}_2(\alpha, r_1, \mathbf{h}) &= (\alpha - h_0) \cdot kE_{y_2}^2 + r_1 \cdot rE_{y_2}^2 \mathbf{h}_2 \end{aligned}$$

$$E_{(x_1, x_2), (y_1, y_2)} = \frac{1}{2} \begin{pmatrix} 0 & \mathbf{rD}_{x_1, y_1}^{1\top} & \mathbf{rD}_{x_2, y_2}^{2\top} \\ -\mathbf{sD}_{x_1, y_1}^1 & \mathbf{0}_{s_1, r_1} & \mathbf{0}_{s_1, r_2} \\ -\mathbf{sD}_{x_2, y_2}^2 & \mathbf{0}_{s_2, r_1} & \mathbf{0}_{s_2, r_2} \end{pmatrix}$$

$$\text{Emb}(pe^1) \wedge_{\text{pair}} \text{Emb}(pe^2) = \begin{cases} \text{Enc1}((x_1, x_2)) = (c_0, \mathbf{c}_1, \mathbf{c}_2) \\ \text{Enc2}((y_1, y_2)) = (k_0, \mathbf{k}_1, k_2, \mathbf{k}_3) \\ \text{Pair}((x_1, x_2), (y_1, y_2)) = E_{(x_1, x_2), (y_1, y_2)} \end{cases}$$

where $\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2)$ and

$$\begin{aligned} c_0(s_0, \mathbf{h}) &= s_0 & k_0(\alpha, (r_1, r_2, r_3), \mathbf{h}) &= r_1 \\ c_1(s_0, \mathbf{h}) &= s_0 \cdot \mathbf{sE}_{x_1}^1 \mathbf{h}_1 & \mathbf{k}_1(\alpha, (r_1, r_2, r_3), \mathbf{h}) &= r_3 \cdot \mathbf{kE}_{y_1}^1 + r_1 \cdot \mathbf{rE}_{y_1}^1 \mathbf{h}_1 \\ c_2(s_0, \mathbf{h}) &= s_0 \cdot \mathbf{sE}_{x_2}^2 \mathbf{h}_2 & k_2(\alpha, (r_1, r_2, r_3), \mathbf{h}) &= r_2 \\ & & \mathbf{k}_3(\alpha, (r_1, r_2, r_3), \mathbf{h}) &= (\alpha - r_3) \cdot \mathbf{kE}_{y_2}^2 + r_2 \cdot \mathbf{rE}_{y_2}^2 \mathbf{h}_2 \end{aligned}$$

$$E_{(x_1, x_2), (y_1, y_2)} = \begin{pmatrix} 0 & \mathbf{rD}_{x_1, y_1}^{1\top} & 0 & \mathbf{rD}_{x_2, y_2}^{2\top} \\ -\mathbf{sD}_{x_1, y_1}^1 & \mathbf{0}_{s_1, r_1} & \mathbf{0}_{s_1} & \mathbf{0}_{s_1, r_2} \\ \mathbf{0}_{s_2} & \mathbf{0}_{s_2, r_1} & -\mathbf{sD}_{x_2, y_2}^2 & \mathbf{0}_{s_2, r_2} \end{pmatrix}$$

The resulting pair encodings are different. The first one (result of our conjunction) does not introduce new random variables and does not increase the number of pairings for decryption. On the other hand, the second conjunction adds new random variables to key generation and increases the number of pairings needed during decryption. This overhead will be amplified if nested conjunctions are used. We include a detailed comparison between the dual transformations in the full version of this paper.

6 Constructions

We provide new instances of predicate encodings to achieve predicate encryption schemes with new properties or better performance.

6.1 Combining Predicates

Dual-Policy ABE. Dual-Policy Attribute Based Encryption [9, 10] has already been considered in the *pair encodings* framework. It combines KP-ABE and CP-ABE into a single construction that simultaneously allows two access control mechanisms. The main advantage is the possibility of considering policies over *objective* attributes (associated to data) and policies over *subjective* attributes (associated to user credentials) at the same time.

Our combinations of predicate encodings allow us to create predicate encryption constructions for Dual-Policy ABE in the framework of pair encodings and tag-based encodings. In particular, given an arbitrary predicate encoding for $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$, applying Theorems 7 and 5 we get an encoding for DP-ABE, i.e., for the predicate $P^* : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{Y} \times \mathcal{X}) \rightarrow \{0, 1\}$ defined as

$$P^*((x, y), (y', x')) = 1 \text{ iff } P(x, y) \wedge P(y', x')$$

Revocation. Another application of our combinations is predicate encryption with revocation, by combining a *boolean formula predicate encoding* with a *broadcast encryption predicate encoding*. The former is used to encode the actual policy of the scheme, while the latter takes care of revocation.

Broadcast encryption has been considered in the literature to approach revocation [19, 23, 30]. In broadcast encryption, a broadcasting authority encrypts a message in such a way that only authorized users will be able to decrypt it. This can be expressed with the predicate $P(\mathbf{x}, i) = 1$ if and only if $\mathbf{x}_i = 1$, where $\mathbf{x} \in \mathcal{X} = \{0, 1\}^n$ and $i \in \mathcal{Y} = [n]$. A drawback is that the number of users in the system, n , is polynomial size. Figure 1 shows the performance of predicate encryption built from a predicate encoding that combines boolean formulas with broadcast encryption. The system supports thousands of users in reasonable time.

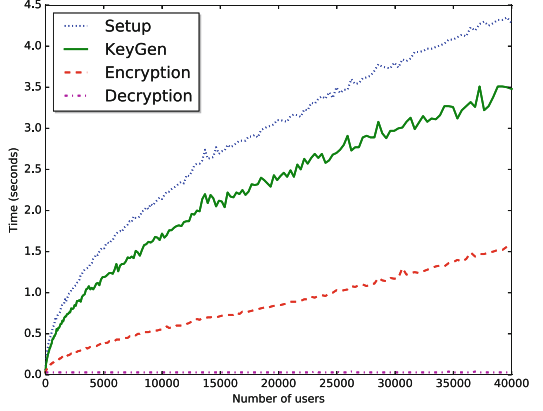


Fig. 1. Scalability of the PE for revocation

6.2 Improved Predicate Encodings

In this section we propose new predicate encodings that are more efficient than some of the encodings proposed previously in [15]. Our encodings are built by applying Theorem 6 to obtain negated encodings and observing that, in some cases, Theorem 3 can be applied to simplify the negated version into a more efficient encoding than the original one. The predicate associated to this new encoding is negated, but if inputs are also negated, the predicate will be equivalent. Figure 2 illustrates this idea. On the left, there is a boolean formula CP-ABE for 4 attributes $\{a, b, c, d\}$. On the right side, secret keys and policies are modified so that the negated version is equivalent. The attribute universe is formed by the *negated attributes*, secret keys are formed by all *negated attributes* do not appear in the original key as normal attributes, policies are negated and expressed in NNF (Negation Normal Form).

Boolean Formulas. In [15], the authors propose two predicate encoding (KP and CP versions) for monotonic boolean formulas. The predicate they consider is a particular case of a Linear Secret Sharing scheme [24]. Let $\mathcal{X} = \{0, 1\}^n$, $\mathcal{Y} = \mathbb{Z}_p^{n \times k}$ for some $n, k \in \mathbb{N}$,

$$P(\mathbf{x}, M) = 1 \text{ iff } \begin{pmatrix} 1 & 0 & \vdots & 1 & 0 \end{pmatrix} \in \overset{\text{row}}{\underset{\text{span}}{\langle M_{\mathbf{x}} \rangle}}$$

$P(\mathbf{x}, M) = 1 \text{ iff } \begin{pmatrix} 1 & 0 & \dots & 1 & 0 \end{pmatrix} \in \text{row}_{\text{span}}(M_{\mathbf{x}})$	
P	\bar{P}
attributes = $\{a, b, c, d\}$	attributes = $\{\bar{a}, \bar{b}, \bar{c}, \bar{d}\}$
$x = (a \wedge c) \vee d$	$x = (\bar{a} \vee \bar{c}) \wedge \bar{d}$
$y = \{a, c\}$	$y = \{\bar{b}, \bar{d}\}$
$P(x, y) = 1 \text{ iff } x(y)$	$\bar{P}(x, y) = 1 \text{ iff } \neg x(y)$

Fig. 2. Equivalent encodings of a policy using P (CP-ABE) on the left and \bar{P} (negated CP-ABE) on the right.

where $M_{\mathbf{x}}$ denotes the matrix M filtered by \mathbf{x} , i.e., $M_{\mathbf{x}}$ includes the i -th row of M iff $x_i = 1$.

It has been shown [28] that for every⁵ monotonic boolean formula f with attributes from \mathcal{X} there exists a matrix $M \in \mathcal{Y}$ such that for every $\mathbf{x} \in \mathcal{X}$, $f(\mathbf{x}) \Leftrightarrow P(\mathbf{x}, M)$. The key-policy predicate encoding from [15] is the following,

$$\mathbf{sE}_{\mathbf{x}} = (\text{diag}(\mathbf{x}) \quad \mathbf{0}_{n, k-1}) \quad \mathbf{rE}_M = \begin{pmatrix} I_n & M_{\{2, \dots, k\}} \end{pmatrix} \quad \mathbf{kE}_M = (M_{\{1\}})$$

where $M_{\{1\}}$ denotes the first column of matrix M , $M_{\{2, \dots, k\}}$ represents the rest of the matrix. We do not include explicit decryption functions \mathbf{sD} and \mathbf{rD} , but they can be computed efficiently by *Gaussian elimination*.

In the above encoding, the number of elements in secret keys and ciphertexts is always maximal, it equals the number of (possibly duplicated) attributes, even for small policies. Furthermore, the maximum number of *and-gates* in a policy must be fixed a priori (because it is related the number of columns in the matrix).

We propose the following improved predicate encoding for (negated) key-policy monotonic boolean formulas, which is an equivalent predicate if instantiated with negated inputs. Let $\mathcal{X} = \{0, 1\}^n$ and $\mathcal{Y} = \mathbb{Z}_p^{n \times k}$,

$$\mathbf{sE}_{\mathbf{x}} = I_n - \text{diag}(\mathbf{x}) \quad \mathbf{rE}_M = M^\top \quad \mathbf{kE}_M = \begin{pmatrix} 1 & 0 & \dots & 1 & 0 \end{pmatrix}^\top$$

In our encoding, the number of columns has been reduced up to half⁶. Furthermore, the size of secret keys is proportional to the complexity of policies. In particular, it is equal to the number of *and-gates* in the policy (or equivalently, the number of *or-gates* in the non-negated version). Note that our improvement also works in the ciphertext-policy case.

In Fig. 3 we present a comparison between our improved encoding for *key-policy monotonic boolean formulas* and the original one. To this end, we generate random boolean formulas for different sizes, starting from a random set of leaf

⁵ Where every attribute appears at most once and the number of *and-gates* is lower than k (one could overcome the one-use restriction by considering duplicated attributes).

⁶ Being half when the bound on the number of *and-gates* is maximal.

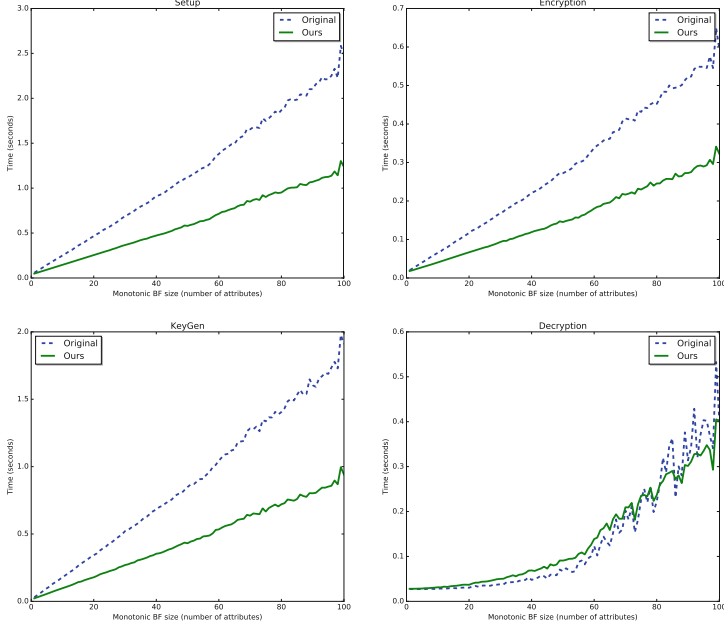


Fig. 3. Improved predicate encoding for boolean formulas vs original encoding

nodes and combining them with boolean operators \vee and \wedge . Our tables report on the average time for each algorithm. Our encoding needs 50% less time than the original algorithms for setup, encryption and key generation. For decryption the performance is similar. All the analyzed schemes were instantiated with the same compiler, therefore all achieve the same level of security (under SXDH assumption). In terms of secret key size, our encoding is smaller in general (in the worst case, when all the gates in the policy are *or-gates*, key sizes are equal).

Arithmetic Span Programs. Chen et al. proposed in [15] a predicate encoding for *Arithmetic Span Programs (ASP)*. That is, an encoding for the predicate P defined as follows. Let $\mathcal{X} = \mathbb{Z}_p^n$, $\mathcal{Y} = \mathbb{Z}_p^{n \times k} \times \mathbb{Z}_p^{n \times k}$, for some $n, k \in \mathbb{N}$; for every $\mathbf{x} \in \mathcal{X}$ and every $(Y, Z) \in \mathcal{Y}$,

$$P(\mathbf{x}, (Y, Z)) = 1 \text{ iff } \begin{pmatrix} 1 & 0 & \vdots & 0 \end{pmatrix} \in \overset{\text{row}}{\text{span}} \langle \text{diag}(\mathbf{x})Y + Z \rangle$$

In [21], Ishai and Wee show how to relate *Arithmetic Span Programs* computations of polynomial functions over a finite field \mathbb{F} , i.e., functions $f : \mathbb{F}^n \rightarrow \mathbb{F}$ that only use addition and multiplication over the field. Therefore, the above predicate can be seen as $f(\mathbf{x}) = 0$, where f is the polynomial function induced by (Y, Z) . Let $\mathcal{X} = \mathbb{Z}_p^n$, $\mathcal{Y} = \mathbb{Z}_p^{n \times k} \times \mathbb{Z}_p^{n \times k}$, the original predicate encoding for arithmetic span programs proposed in [15] is the following:

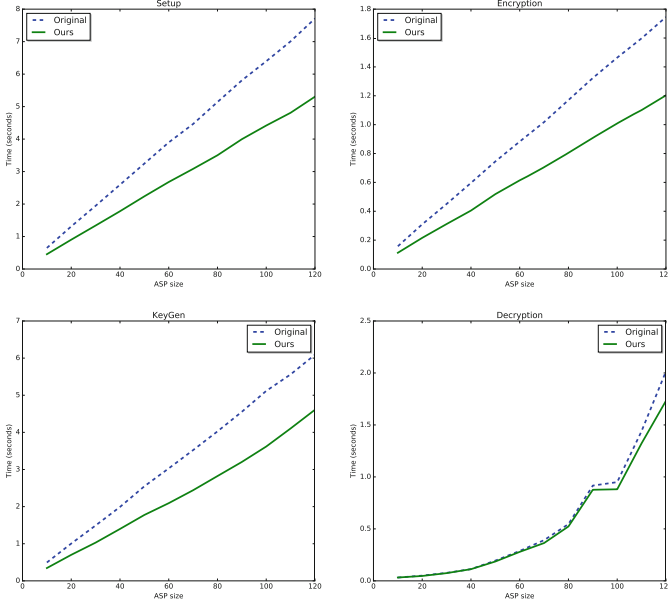


Fig. 4. Improved predicate encoding for ASP vs original encoding

$$\mathbf{sE}_x = \begin{pmatrix} \text{diag}(\mathbf{x}) & I_n & \mathbf{0}_{n,k-1} \end{pmatrix} \quad \mathbf{rE}_{(Y,Z)} = \begin{pmatrix} I_n & \mathbf{0}_{n,n} & Y_{\{2,\dots,l\}} \\ \mathbf{0}_{n,n} & I_n & Z_{\{2,\dots,l\}} \end{pmatrix} \quad \mathbf{kE}_{(Y,Z)} = \begin{pmatrix} Y_{\{1\}} \\ Z_{\{1\}} \end{pmatrix}$$

We present a more efficient encoding for (negated⁷) arithmetic span programs:

$$\mathbf{sE}_x = \begin{pmatrix} \text{diag}(\mathbf{x}) & -I_n \end{pmatrix} \quad \mathbf{rE}_{(Y,Z)} = \begin{pmatrix} Z^\top & Y^\top \end{pmatrix} \quad \mathbf{kE}_{(Y,Z)} = \begin{pmatrix} 1 & 0 & k-1 & 0 \end{pmatrix}^\top$$

Figure 4 shows the performance of our new encoding for KP-ABE for Arithmetic Span Programs compared to the original encoding from [15]. As we expected, our encoding needs 66% of the time required for the original encoding for setup, encryption and key generation. Additionally, secret key size is halved with our encoding.

6.3 Extra Features

In this section we consider new theoretical results that can be proved thanks to our algebraic characterization of α -privacy and can be used to produce new predicate encodings enhanced with extra properties.

⁷ In [21] there is a modification of their algorithm that produces matrices (Y, Z) such that the predicate associated is $f(\mathbf{x}) \neq 0$ (the double negation will cancel out).

Attribute-Hiding for Boolean Formulas. Chen et al. proposed an extension of the compiler in [15] to build weakly attribute-hiding predicate encryption schemes [13, 25]. In a weakly attribute-hiding scheme, the ciphertext attribute x remains secret for unauthorized users, that only learn the fact that their secret keys are not valid. This additional compiler needs to be instantiated with *predicate encodings* satisfying additional properties. The following is a definition from [15].

Definition 6 (Attribute-Hiding Encodings). A (s, r, w) -predicate encoding, (sE, rE, kE, sD, rD) for $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ is attribute-hiding if it verifies the additional requirements:

x -oblivious reconstruction: $sD_{x,y}$ and $rD_{x,y}$ are independent of x .

attribute-hiding: for all $(x, y) \notin P$,

$$w \stackrel{\$}{\leftarrow} \mathbb{Z}_p^w; \text{ return } (sE_x w, rE_y w) \equiv s \stackrel{\$}{\leftarrow} \mathbb{Z}_p^s; r \stackrel{\$}{\leftarrow} \mathbb{Z}_p^r; \text{ return } (s, r)$$

where \equiv denotes equality of distributions.

The following theorem relates the second property with our alternative definition of predicate encodings:

Theorem 11 (Algebraic characterization of attribute-hiding). Let $p \in \mathbb{N}$ be a prime, let $s, r, w \in \mathbb{N}$ and let $S \in \mathbb{Z}_p^{s \times w}$, $R \in \mathbb{Z}_p^{r \times w}$, $k \in \mathbb{Z}_p^r$. The following are equivalent:

- $w \stackrel{\$}{\leftarrow} \mathbb{Z}_p^w; \text{ return } (Sw, Rw) \equiv s \stackrel{\$}{\leftarrow} \mathbb{Z}_p^s; r \stackrel{\$}{\leftarrow} \mathbb{Z}_p^r; \text{ return } (s, r)$
- $\text{rank} \begin{pmatrix} S \\ R \end{pmatrix} = s + r$

Note that for every (s, r, w) -predicate encoding (sE, rE, kE, sD, rD) that is attribute-hiding, there exists an equivalent $(s, 1, w)$ -predicate encoding. This is because rD is independent from x and thus, we can apply our optimization Theorem 3 with matrices $B_y = rD_{x,y}^\top \in \mathbb{Z}_p^{1 \times w}$, $A_x = I_s$, $C = I_w$. Therefore, the class of predicates that can be built from attribute-hiding encodings is included in the class of predicates achieved from $(s, 1, w)$ -predicate encodings.

Further, note that our *disjunction* and *conjunction* combinations for predicate encodings (Theorems 4 and 5 respectively) preserve the notion of attribute-hiding⁸. Exploiting this fact, we propose a Policy-Hiding ABE scheme for non-monotonic boolean formulas expressed in DNF (Disjunctive Normal Form). The inner product can be used to encode conjunctions [25]. More concretely, let $\mathbf{y} \in \{0, 1\}^n \subseteq \mathbb{Z}_p^n$. We establish that the i -th attribute a_i appears in a secret key for \mathbf{y} iff $\mathbf{y}_i = 1$. Let $S, \bar{S} \subseteq \{a_i\}_{i=1}^n$ be sets such that $S \cap \bar{S} = \emptyset$,⁹

$$\bigwedge_{a \in S} a \wedge \bigwedge_{a \in \bar{S}} \bar{a} \stackrel{9}{\Leftrightarrow} \mathbf{x}^\top \mathbf{y} = |S| \quad \text{where } \forall i \in [n], \mathbf{x}_i = \begin{cases} 1 & \text{if } a_i \in S \\ -1 & \text{if } a_i \in \bar{S} \\ 0 & \text{otherwise} \end{cases}$$

⁸ Conjunction also preserves x -oblivious reconstruction, while disjunction does not.

⁹ This equivalence holds when $|S| < p$, but in practice p is a large prime.

Note that the ZIPE predicate encoding from [15, Appendix A.1] can be modified into an attribute-hiding encoding for the predicate $P((\mathbf{x}, \gamma), \mathbf{y}) = 1$ iff $\mathbf{x}^\top \mathbf{y} = \gamma$.

Let $\mathcal{U} = \{a, b, c\}$ be the set of attributes. We consider the predicate encoding for monotonic boolean formulas from [15]. Let $\mathcal{X} = \{0, 1\}^3, \mathcal{Y} = \mathbb{Z}_p^{3 \times 2}$,

$$\mathbf{sE}_x = (\text{diag}(\mathbf{x}) \quad \mathbf{0}_{3,2}) \quad \mathbf{rE}_M = (I_3 \quad M_{\{2\}}) \quad \mathbf{kE}_M = (M_{\{1\}})$$

The following is the encoding of a key for the formula $(a \vee c) \wedge b$, enhanced for delegation according to Theorem 12 (with $k = 1$),

$$\mathbf{rE}_M = \left(\begin{array}{cccc|c} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right) \quad \mathbf{kE}_M = \left(\begin{array}{c} 1 \\ 0 \\ 1 \\ 0 \end{array} \right)$$

Let's assume we want to weaken this key to one for the formula $a \wedge b \wedge c$. Note that in this case we want to make an update of the matrix M :

$$M = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \text{ encodes } (a \vee c) \wedge b \quad M' = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \text{ encodes } a \wedge b \wedge c$$

It can be done by multiplying \mathbf{rE}_M from the left by A

$$\mathbf{rE}'_M = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{rE}_M} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{kE}'_M = A \cdot \mathbf{kE}_M = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Fig. 5. Example of delegation of keys for monotonic boolean formulas. Since A is a linear function, it can be computed in the exponent from the given key.

Therefore, with a disjunction of k predicate encodings like the former we can encode boolean formulas that have at most k disjuncts. Note that the resulting encoding is *attribute-hiding* but it is not *x-oblivious*. However, without the knowledge of the policy x , one can guess for the disjunct his secret key satisfies (if any). In this way, a valid key will be enough to decrypt after at most k decryption tries (one for every disjunct).

Delegation. Delegation of keys is a desirable property for every predicate encryption scheme. Roughly, it allows the owner of a secret key to weaken his key creating a new one that is less powerful than the original one. This property can be used to achieve *forward secrecy* (see [14] for an application to ABE that supports delegation), where past sessions are protected from the compromise of future secret keys. More formally, we say that a predicate $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ supports delegation if there is a partial ordering \preceq on \mathcal{Y} such that for every $x \in \mathcal{X}$, if $P(x, y) = 1$ and $(y \preceq y')$, then $P(x, y') = 1$.

Delegation has been considered in [15] as the property of some predicate encodings. We propose a generic method to convert any predicate encoding into one supporting delegation.

Theorem 12 (Delegation). *For every (s, r, w) -predicate encoding (sE, rE, kE, sD, rD) for $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$, for every $k \in \mathbb{N}$, $(sE', rE', kE', sD', rD')$ defined below is a valid $(s, r + k, w + k)$ -predicate encoding for P .*

$$\begin{aligned} sE'_x &= (sE_x \quad \mathbf{0}_{s,k}) & rE'_y &= \begin{pmatrix} rE_y & \mathbf{0}_{r,k} \\ \mathbf{0}_{k,w} & I_k \end{pmatrix} & kE'_y &= \begin{pmatrix} kE_y \\ \mathbf{0}_k \end{pmatrix} \\ sD'_{x,y} &= sD_{x,y} & rD'_{x,y} &= \begin{pmatrix} rD_{x,y} \\ \mathbf{0}_k \end{pmatrix} \end{aligned}$$

The additional set of not-null rows in rE'_y can be used to weaken the linear span of rE_y , what directly modifies the predicate. In particular, this method works really well for monotonic boolean formulas (see Fig. 5 for an example).

Acknowledgements. The work presented here was supported by projects S2013/ICE-2731 N-GREENS Software-CM, ONR Grants N000141210914 and N000141512750.

A Proofs from Main Body

Proof (Of Theorem 2). The proof follows from Theorem 1 and the observation that reconstructability of predicate encodings is equivalent to Pred , while privacy of predicate encodings is equivalent to $\neg \text{Pred}$. \square

Proof (Of Theorem 3). To see correctness of the new encoding, note that for all $(x, y) \in P$, since

$$sD_{x,y}^\top \in \text{row}_{\text{span}} \langle A_x \rangle \wedge rD_{x,y}^\top \in \text{row}_{\text{span}} \langle B_y \rangle$$

there exist $sD'_{x,y}{}^\top$ and $rD'_{x,y}{}^\top$ such that

$$sD_{x,y}^\top = sD'_{x,y}{}^\top A_x \wedge rD_{x,y}^\top = rD'_{x,y}{}^\top B_y$$

Therefore,

$$\begin{aligned} sD'_{x,y}{}^\top (A_x sE_x C) &= (sD_{x,y}^\top sE_x) C = (rD_{x,y}^\top rE_y) C = rD'_{x,y}{}^\top (B_y rE_y C) \\ rD'_{x,y}{}^\top (B_y kE_y) &= rD_{x,y}^\top kE_y = 1 \end{aligned}$$

To see privacy, note that for every $(x, y) \notin P$, there exists $\mathbf{w} \in \text{col}_{\text{span}} \langle C \rangle$ such that $sE_x \mathbf{w} = \mathbf{0}_s \wedge rE_y \mathbf{w} = kE_y$. Therefore, there also exists $\mathbf{w}' \in \mathbb{Z}_p^{w'}$ such that $\mathbf{w} = C\mathbf{w}'$. Note that,

$$\begin{aligned} sE'_x \mathbf{w}' &= (A_x sE_x C) \mathbf{w}' = A_x sE_x \mathbf{w} = A_x \mathbf{0}_s = \mathbf{0}_s \\ rE'_y \mathbf{w}' &= (B_y rE_y C) \mathbf{w}' = B_y rE_y \mathbf{w} = B_y kE_y = kE'_y \end{aligned}$$

so algebraic privacy is satisfied. \square

Proof (Of Theorem 4). Reconstructability can be seen by a simple check based on the reconstructability of the original encodings.

To see privacy, note that $P_1(x_1, y_1) \vee P_2(x_2, y_2) = 0$ implies $P_1(x_1, y_1) = 0$ and $P_2(x_2, y_2) = 0$ implies. Let \mathbf{w}_1 and \mathbf{w}_2 be witnesses of privacy of predicate encodings 1 and 2 respectively. It is easy to check that $\mathbf{w}^\top = (\mathbf{w}_1^\top \ \mathbf{w}_2^\top)$ is a witness of privacy of the transformed encoding. \square

Proof (Of Theorem 5). A simple check shows reconstructability. To see privacy, $P_1(x_1, y_1) \wedge P_2(x_2, y_2) = 0$ implies $P_1(x_1, y_1) = 0$ or $P_2(x_2, y_2) = 0$. If the first holds, let \mathbf{w}_1 be a witness of privacy of the first encoding. Then, $\mathbf{w}^\top = (2\mathbf{w}_1^\top \ \mathbf{0}_{w_2}^\top \ -1)$ is a witness of the algebraic privacy of the transformed encoding. If the second holds, let \mathbf{w}_2 be a witness of privacy of the second encoding. A valid witness now is $\mathbf{w}^\top = (\mathbf{0}_{w_2}^\top \ 2\mathbf{w}_2^\top \ 1)$. \square

Proof (Of Theorem 6). It is not difficult to check reconstructability. Privacy holds because when $P(x, y) = 1$, we can define $\mathbf{w}^\top = (-sD_{x,y}^\top \ -sD_{x,y}^\top sE_x \ rD_{x,y}^\top)$ which can be checked to be a witness of the algebraic privacy of the transformed predicate encoding. \square

Proof (Of Theorem 7). A simple check is enough to verify reconstructability. For privacy, note that when $P'(y, x) = 0$, we have $P(x, y) = 0$. Let \mathbf{w} be a witness of the algebraic privacy of the original encoding. Now, $\mathbf{w}'^\top = (-\mathbf{w}^\top \ 1)$ is a witness of the dual predicate encoding. \square

Proof (Of Theorem 8). The proof follows directly from the following lemma and the observation that (i) is equivalent to \mathbf{h} -hiding, while (iii) is non-reconstructability (take $A = C$ and $B = K$). \square

Lemma 1. Let $A \in \mathbb{Z}_p^{m \times n}$ and $B \in \mathbb{Z}_p^{l \times n}$ be matrices. Let $C \in \mathbb{Z}_p^{(m+l) \times n}$ be the concatenation of A and B by rows. The following three statements are equivalent:

$$(i) \ \forall \mathbf{a} \in \mathbb{Z}_p^m, \forall \mathbf{b} \in \mathbb{Z}_p^l, \ \Pr_{\mathbf{x} \leftarrow \mathbb{Z}_p^n} [A\mathbf{x} = \mathbf{a} \mid B\mathbf{x} = \mathbf{b}] = \Pr_{\mathbf{x} \leftarrow \mathbb{Z}_p^n} [A\mathbf{x} = \mathbf{a}]$$

$$(ii) \ \text{rank}(C) = \text{rank}(A) + \text{rank}(B)$$

$$(iii) \ \forall \mathbf{a} \in \mathbb{Z}_p^m, \forall \mathbf{b} \in \mathbb{Z}_p^l, \ \mathbf{a}^\top A \neq \mathbf{b}^\top B \ \vee \ \mathbf{a}^\top A = \mathbf{0}_n^\top$$

Proof (Of the Lemma). Note that (i) holds for every $\mathbf{a} \in \mathbb{Z}_p^m, \mathbf{b} \in \mathbb{Z}_p^l$ such that $A\mathbf{x} = \mathbf{a}$ or $B\mathbf{x} = \mathbf{b}$ have no solution. Let $\mathbf{a} \in \mathbb{Z}_p^m, \mathbf{b} \in \mathbb{Z}_p^l$ be such that the systems $A\mathbf{x} = \mathbf{a}$ and $B\mathbf{x} = \mathbf{b}$ have individually at least one solution (note that such \mathbf{a} and \mathbf{b} always exist). We define the sets $\Omega_A = \{\mathbf{x} \in \mathbb{Z}_p^n : A\mathbf{x} = \mathbf{a}\}$, $\Omega_B = \{\mathbf{x} \in \mathbb{Z}_p^n : B\mathbf{x} = \mathbf{b}\}$, $\Omega_{AB} = \{\mathbf{x} \in \mathbb{Z}_p^n : A\mathbf{x} = \mathbf{a} \wedge B\mathbf{x} = \mathbf{b}\}$. By the Rouché-Capelli Theorem,

$$|\Omega_A| = p^{n-\text{rank}(A)} \quad |\Omega_B| = p^{n-\text{rank}(B)} \quad |\Omega_{AB}| = p^{n-\text{rank}(C)}$$

Note that (i) can be expressed as $\frac{|\Omega_{AB}|}{p^n} = \frac{|\Omega_A|}{p^n} \cdot \frac{|\Omega_B|}{p^n}$ which is equivalent to the equation $p^n \cdot |\Omega_{AB}| = |\Omega_A| \cdot |\Omega_B|$, and therefore, $p^n \cdot p^{n-\text{rank}(C)} = p^{n-\text{rank}(A)} \cdot p^{n-\text{rank}(B)}$ if and only if $\text{rank}(C) = \text{rank}(A) + \text{rank}(B)$ which is (ii).

Now, note that $\text{rank}(C) = \text{rank}(A) + \text{rank}(B)$ if and only if there is not a non-zero linear combination of rows of A that can be expressed as a linear combination of rows of B , which is equivalent to statement (iii). \square

Proof (Of Theorem 9). According to our Theorem 2, the partial encoding $(\mathbf{sE}', \mathbf{rE}', \mathbf{kE}')$ induces a predicate encoding for the predicate $\text{Pred}(x, y) = 1$ iff $\exists \mathbf{s} \in \mathbb{Z}_p^c, \mathbf{r} \in \mathbb{Z}_p^1$ s.t. $\mathbf{s}^\top \mathbf{sE}'_x = \mathbf{r} \cdot \mathbf{rE}'_y$ and $\mathbf{r} \cdot \mathbf{kE}'_y = 1$, or equivalently, $\exists \mathbf{s} \in \mathbb{Z}_p^c$ s.t. $\mathbf{s}^\top \mathbf{cE}_x = \mathbf{kE}_y$, which is equivalent to the reconstructability of the tag-based encoding $(\mathbf{cE}, \mathbf{kE})$. According to Theorem 8 it is also equivalent to the predicate P . \square

Proof (Of Theorem 10). Verifying correctness of the pair encoding is a simple check. For perfect security we need to check that, when $(x, y) \notin P$, the following two distributions are identical:

$$\begin{aligned} \alpha, s_0 \xleftarrow{\$} \mathbb{Z}_p; r_1 \xleftarrow{\$} \mathbb{Z}_p^*; \mathbf{h} \xleftarrow{\$} \mathbb{Z}_p^w; \text{return } (s_0, s_0 \cdot \mathbf{sE}_x \mathbf{h}, r_1, r_1 \cdot \mathbf{rE}_y \mathbf{h}) &\equiv \\ s_0 \xleftarrow{\$} \mathbb{Z}_p; r_1 \xleftarrow{\$} \mathbb{Z}_p^*; \mathbf{h} \xleftarrow{\$} \mathbb{Z}_p^w; \text{return } (s_0, s_0 \cdot \mathbf{sE}_x \mathbf{h}, r_1, r_1 \cdot \mathbf{rE}_y \mathbf{h} + \alpha \cdot \mathbf{kE}_y) & \end{aligned}$$

Since both distributions provide s_0 and r_1 in the clear, the above checking is equivalent to the following:

$$\begin{aligned} \mathbf{h} \xleftarrow{\$} \mathbb{Z}_p^w; \text{return } (\mathbf{sE}_x \mathbf{h}, \mathbf{rE}_y \mathbf{h}) &\equiv \\ \alpha \xleftarrow{\$} \mathbb{Z}_p; r_1 \xleftarrow{\$} \mathbb{Z}_p^*; \mathbf{h} \xleftarrow{\$} \mathbb{Z}_p^w; \text{return } (\mathbf{sE}_x \mathbf{h}, \mathbf{rE}_y \mathbf{h} + \alpha/r_1 \cdot \mathbf{kE}_y) & \end{aligned}$$

but those distributions are identical due to the α -privacy of the predicate encoding¹⁰. \square

Proof (Of Theorem 11). Given $(\mathbf{s}, \mathbf{r}) \in \mathbb{Z}_p^s \times \mathbb{Z}_p^r$, we define $\Omega_{\mathbf{s}, \mathbf{r}} = \{\mathbf{w} \in \mathbb{Z}_p^w : \mathbf{S}\mathbf{w} = \mathbf{s} \wedge \mathbf{R}\mathbf{w} = \mathbf{r}\}$. The condition on the second bullet holds iff $w - s - r \geq 0$ and the cardinality of $\Omega_{\mathbf{s}, \mathbf{r}}$ is p^{w-s-r} . Additionally, $|\Omega_{\mathbf{s}, \mathbf{r}}|$ is independent from \mathbf{r} and \mathbf{s} iff the two distributions from the first bullet are identical. \square

Proof (Of Theorem 12). Correctness can be easily checked. For privacy, let $(x, y) \notin P$ and let $\mathbf{w} \in \mathbb{Z}_p^w$ be such that $\mathbf{sE}_x \mathbf{w} = \mathbf{0}_s$ and $\mathbf{rE}_y \mathbf{w} = \mathbf{kE}_y$. Note that $\mathbf{w}'^\top = (\mathbf{w}^\top \quad \mathbf{0}_k^\top)$ is a witness of privacy for $(\mathbf{sE}', \mathbf{rE}', \mathbf{kE}', \mathbf{sD}', \mathbf{rD}')$. \square

References

1. Agrawal, S., Chase, M.: A study of pair encodings: predicate encryption in prime order groups. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9563, pp. 259–288. Springer, Heidelberg (2016). doi:[10.1007/978-3-662-49099-0_10](https://doi.org/10.1007/978-3-662-49099-0_10)

¹⁰ Note that $\alpha \xleftarrow{\$} \mathbb{Z}_p, r_1 \xleftarrow{\$} \mathbb{Z}_p^*$ and therefore, α/r_1 distributes uniformly over \mathbb{Z}_p , so we can apply the α -privacy property from the predicate encoding.

2. Agrawal, S., Chase, M.: Simplifying design and analysis of complex predicate encryption schemes. Cryptology ePrint Archive, Report 2017/233 (2017). EUROCRYPT (2017)
3. Akinyele, J.A., Lehmann, C.U., Green, M.D., Pagano, M.W., Peterson, Z.N.J., Rubin, A.D.: Self-protecting electronic medical records using attribute-based encryption. Cryptology ePrint Archive, Report 2010/565 (2010). <http://eprint.iacr.org/2010/565>
4. Applebaum, B., Arkis, B., Raykov, P., Vasudevan, P.N.: Conditional disclosure of secrets: amplification, closure, amortization, lower-bounds, and separations. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 24, p. 38 (2017)
5. Aranha, D.F., Gouvêa, C.P.L.: RELIC is an Efficient Library for Cryptography. <https://github.com/relic-toolkit/relic>
6. Attrapadung, N.: Dual system encryption via doubly selective security: framework, fully secure functional encryption for regular languages, and more. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577. Springer, Heidelberg (2014). doi:[10.1007/978-3-642-55220-5_31](https://doi.org/10.1007/978-3-642-55220-5_31)
7. Attrapadung, N.: Dual system encryption framework in prime-order groups. Cryptology ePrint Archive, Report 2015/390 (2015). <http://eprint.iacr.org/2015/390>
8. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute-based encryption. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 248–265. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-03298-1_16](https://doi.org/10.1007/978-3-642-03298-1_16)
9. Attrapadung, N., Imai, H.: Dual-policy attribute based encryption. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 168–185. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-01957-9_11](https://doi.org/10.1007/978-3-642-01957-9_11)
10. Attrapadung, N., Yamada, S.: Duality in ABE: converting attribute based encryption for dual predicate and dual policy via computational encodings. Cryptology ePrint Archive, Report 2015/157 (2015). <http://eprint.iacr.org/2015/157>
11. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. Cryptology ePrint Archive, Report 2005/133 (2005). <http://eprint.iacr.org/2005/133>
12. Beimel, A.: Secret-sharing schemes: a survey. In: Chee, Y.M., Guo, Z., Ling, S., Shao, F., Tang, Y., Wang, H., Xing, C. (eds.) IWCC 2011. LNCS, vol. 6639, pp. 11–46. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-20901-7_2](https://doi.org/10.1007/978-3-642-20901-7_2)
13. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-70936-7_29](https://doi.org/10.1007/978-3-540-70936-7_29)
14. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. J. Cryptol. **20**(3), 265–294 (2007)
15. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (2015). doi:[10.1007/978-3-662-46803-6_20](https://doi.org/10.1007/978-3-662-46803-6_20)
16. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-40084-1_25](https://doi.org/10.1007/978-3-642-40084-1_25)
17. Chen, J., Wee, H.: Dual system groups and its applications – compact hibe and more. Cryptology ePrint Archive, Report 2014/265 (2014). <http://eprint.iacr.org/2014/265>
18. Dinh, T.T.A., Datta, A.: Streamforce: outsourcing access control enforcement for stream data to the clouds. In: Fourth ACM Conference on Data and Application Security and Privacy, CODASPY 2014, San Antonio, TX, USA, 03–05 March 2014, pp. 13–24 (2014)

19. Garg, S., Kumarasubramanian, A., Sahai, A., Waters, B.: Building efficient fully collusion-resilient traitor tracing and revocation schemes. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 10, pp. 121–130. ACM Press, October 2010
20. Gay, R., Kerenidis, I., Wee, H.: Communication complexity of conditional disclosure of secrets and attribute-based encryption. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 485–502. Springer, Heidelberg (2015). doi:[10.1007/978-3-662-48000-7_24](https://doi.org/10.1007/978-3-662-48000-7_24)
21. Ishai, Y., Wee, H.: Partial garbling schemes and their applications. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8572, pp. 650–662. Springer, Heidelberg (2014). doi:[10.1007/978-3-662-43948-7_54](https://doi.org/10.1007/978-3-662-43948-7_54)
22. Guo, F., Kim, J., Susilo, W., Au, M.H.: A tag based encoding: an efficient encoding for predicate encoding in prime order groups. Cryptology ePrint Archive, Report 2016/655 (2016). <http://eprint.iacr.org/2016/655>
23. Junod, P., Karlov, A.: An efficient public-key attribute-based broadcast encryption scheme allowing arbitrary access policies. In: Proceedings of the Tenth Annual ACM Workshop on Digital Rights Management, DRM 2010, pp. 13–24. ACM, New York (2010)
24. Karchmer, M., Wigderson, A.: On span programs. In: Proceedings of the 8th IEEE Structure in Complexity Theory, pp. 102–111. IEEE Computer Society Press (1993)
25. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-78967-3_9](https://doi.org/10.1007/978-3-540-78967-3_9)
26. Lewko, A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 318–335. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-29011-4_20](https://doi.org/10.1007/978-3-642-29011-4_20)
27. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-11799-2_27](https://doi.org/10.1007/978-3-642-11799-2_27)
28. Lewko, A.B., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-20465-4_31](https://doi.org/10.1007/978-3-642-20465-4_31)
29. Lewko, A.B., Waters, B.: New proof methods for attribute-based encryption: achieving full security through selective techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-32009-5_12](https://doi.org/10.1007/978-3-642-32009-5_12)
30. Liu, Z., Wong, D.S.: Practical ciphertext-policy attribute-based encryption: traitor tracing, revocation, and large universe. In: Malkin, T., Kolesnikov, V., Lewko, A.B., Polychronakis, M. (eds.) ACNS 2015. LNCS, vol. 9092, pp. 127–146. Springer, Cham (2015). doi:[10.1007/978-3-319-28166-7_7](https://doi.org/10.1007/978-3-319-28166-7_7)
31. Lubicz, D., Sirvent, T.: Attribute-based broadcast encryption scheme made efficient. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 325–342. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-68164-9_22](https://doi.org/10.1007/978-3-540-68164-9_22)
32. Sahai, A., Seyalioglu, H., Waters, B.: Dynamic credentials and ciphertext delegation for attribute-based encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 199–217. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-32009-5_13](https://doi.org/10.1007/978-3-642-32009-5_13)

33. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). doi:[10.1007/11426639_27](https://doi.org/10.1007/11426639_27)
34. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). doi:[10.1007/3-540-39568-7_5](https://doi.org/10.1007/3-540-39568-7_5)
35. Wang, F., Mickens, J., Zeldovich, N., Vaikuntanathan, V.: Sieve: cryptographically enforced access control for user data in untrusted clouds. In: 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), pp. 611–626, Santa Clara, CA. USENIX Association, March 2016
36. Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-03356-8_36](https://doi.org/10.1007/978-3-642-03356-8_36)
37. Wee, H.: Dual system encryption via predicate encodings. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 616–637. Springer, Heidelberg (2014). doi:[10.1007/978-3-642-54242-8_26](https://doi.org/10.1007/978-3-642-54242-8_26)

Advances in Cryptology – CRYPTO 2017

37th Annual International Cryptology Conference, Santa
Barbara, CA, USA, August 20–24, 2017, Proceedings,

Part I

Katz, J.; Shacham, H. (Eds.)

2017, XV, 793 p. 120 illus., Softcover

ISBN: 978-3-319-63687-0