Simplifying Design and Analysis of Complex Predicate Encryption Schemes

Shashank Agrawal[*] Melissa Chase[†]

Abstract

Wee (TCC'14) and Attrapadung (Eurocrypt'14) introduced predicate and pair encodings, respectively, as a simple way to construct and analyze attribute-based encryption schemes, or more generally predicate encryption. However, many schemes do not satisfy the simple information theoretic property proposed in those works, and thus require much more complicated analysis. In this paper, we propose a new simple property for pair encodings called *symbolic* security. Proofs that pair encodings satisfy this property are concise and easy to verify. We show that this property is inherently tied to the security of predicate encryption schemes by arguing that any scheme which is not trivially broken must satisfy it. Then we use this property to discuss several ways to convert between pair encodings to obtain encryption schemes with different properties like small ciphertexts or keys. Finally, we show that any pair encoding satisfying our new property can be used to construct a fully secure predicate encryption scheme. The resulting schemes are secure under a new q-type assumption which we show follows from several of the assumptions used to construct such schemes in previous work.

[*] Shashank Agrawal is a Research Scientist at Visa Research. He can be reached at `shaagraw@visa.com`. Part of this work was done when he was a graduate student at the University of Illinois, Urbana-Champaign, supported by NSF CNS 12-28856 and Andrew & Shana Laursen fellowship, and then an intern at Microsoft Research.

[†] Melissa Chase is a Researcher at Microsoft Research. She can be reached at `melissa@microsoft.com`.

Contents

1 Introduction

Traditional public key encryption allows an encryptor to use a public key to encrypt a message so that the owner of the corresponding secret key can decrypt. In 2005, Sahai and Waters [SW05] introduced the concept of attribute-based encryption, in which who can decrypt is determined by some more complex attributes of the decryptor and the message. Of course this is only meaningful if there is some party that can determine the attributes of the decryption, thus the basic model assumes a trusted party who publishes parameters used in encryption, and who issues decryption keys to users based on their attributes; given such a key, a user should be able to decrypt any ciphertext which is compatible with his attributes. The initial result considered a simple threshold functionality: every ciphertext was encrypted with a set of attributes, and a user could decrypt if they possessed sufficiently many of those attributes. This was then generalized to key-policy ABE [GPSW06], in which the user's key specifies a policy determining what attributes must be present in the ciphertext in order for that user to be able to decrypt, and ciphertext-policy ABE [BSW07], which is the natural opposite in that the user's key corresponds to a list of attributes and ciphertexts are encrypted with a policy which determines which attributes the user must have to decrypt.

Since then the field of ABE has grown dramatically. There has been work which extends the type of policies that can be considered, for example to non-monotone formulas [OSW07], or even regular languages [Wat12]. There has also been work which improves the efficiency of ABE in various dimensions, for example considering schemes with very short (e.g. constant size) ciphertexts or keys [ALdP11, YAHK14], or schemes with very short parameters (again constant-size) which still support attributes from an unbounded space [LW11b, OT12, RW13]. There has been work on distributing the job of the authority across multiple entities [Cha07, LW11a], on updating ciphertexts [SSW12], or hiding the key and/or ciphertext attributes [BW07, KSW08, SSW09, BRS13], and many other interesting directions.[1]

One weakness in much of the early work is that the schemes presented were only shown to satisfy a weak notion of security called *selective security*. Selective security essentially only guarantees security for an adversary who chooses which type of ciphertext to attack (i.e. the attributes/policy for the ciphertext) without seeing the system parameters, any ciphertexts, or any decryption keys. Thus it was a major breakthrough when Waters introduced the dual-system encryption technique [Wat09], paving the way for schemes which satisfied the natural definition, in which the adversary may choose what type of ciphertext to attack adaptively based on any of the other information it sees while interacting with the system. Since then there has been a lot of work focused on obtaining the results above under this more natural security definition, which is usually referred to as *full security*.

One of the main downsides of this process, however, is that while most of the original constructions were simple and intuitive, many of these new constructions are significantly more complex. Also many of the first fully secure schemes relied on composite-order pairing groups, which while conceptually simpler are not really usable in practice [Gui13]. The effort to move these results to be based on standard prime-order pairing groups has added even more complexity [Fre10, Lew12, HHH$^+$14]. As a result, the intuition for the resulting constructions is often difficult to follow, and the security analysis for these schemes is much more involved, so much so that even *verifying* the security proof is often very time consuming.

Two recent works by Wee and Attrapadung [Wee14, Att14a] set out to simplify the process of

[1] There has also been a very interesting line of work which uses indistinguishability obfuscation or multi-linear maps to construct ABE for circuits [GGH$^+$13, GGHZ16], and a lot of progress on building ABE schemes from lattices [Boy13, GVW13], although achieving the natural full security notion there still requires complexity leveraging. Here, we focus on pairing based constructions as to date they provide the best efficiency and security guarantees.

designing and analyzing fully secure ABE schemes. They proposed a simple building block, called a predicate/pair encoding, which essentially considers what happens in the exponent of a single key and a single ciphertext. They proposed an information theoretic security property, which considers the distributions of these values, again only considering a single key and ciphertext, and showed that from any pair encoding scheme which satisfies this property one can construct a fully secure ABE scheme. The initial works proposed only composite-order group schemes; later works [CGW15, AC16, Att16] have updated these results to prime-order groups.

These results led to very simple, intuitive, and easy to analyze constructions for several basic types of ABE schemes, that worked in efficient prime order groups, and were based on simple assumptions like DLIN or SXDH. However, there are many types of ABE schemes for which we do not know how to construct this type of pair encoding. And in fact there are many types of ABE which we do not know how to construct under simple assumptions using any approach, like ABE with short ciphertexts, or with large universe, or where an attribute can be used any number of times in a policy, etc.

To address this problem, Attrapadung [Att14a] also proposed a different security notion for pair encodings, and showed that under this notion one could construct pair encodings for many more types of ABEs, and that this notion was sufficient to produce secure constructions under more complex q-type assumptions. However, proving that a pair encoding scheme satisfies the new security notion is again a challenging task. This property involves elements in bilinear groups rather than just the exponent, and it is no longer information-theoretic, so that it must be proved via reduction to a different q-type assumption for every encoding. These reductions are very complex, and again verifying the security becomes a matter of studying several pages of proof (9 pages for predicate encryption for regular languages, for instance), providing relatively little intuition for why the scheme is secure.

1.1 Our Contributions

Our goal in this work is to simplify the process of designing and analyzing ABE schemes for those types of ABEs which we only know how to construct from q-type assumptions. Towards this, we introduce a very different kind of security property for pair encodings that completely does away with any kind of distributions, and show that it is a very powerful and natural property through a series of results. We believe it provides a new perspective for looking at the security of predicate encryption schemes.

A pair encoding scheme, as defined by Attrapadung [Att14a], gives a way to encode the two inputs x and y to a predicate into polynomials of a simple structure. These polynomials have three types of variables: common variables shared by the encodings of x and y, and variables specific to the encoding of x and to that of y.

A new property for pair encodings. We present a new security property for pair encodings that essentially requires one to describe a mapping from the variables in the encoding to matrices and vectors. Once a mapping is specified, verifying that the property holds is just a matter of checking if the polynomials in the encoding evaluate to 0 when the variables are substituted.[2] Thus verification is much easier compared to any property known before, since they all require checking whether certain distributions are (pefectly, statistically or computationally) indistinguishable. We call our new property the *symbolic property* (Sym-Prop) since verification only involves symbolic manipulation.

[2] The trivial case is ruled out because we also require that the vectors corresponding to two special variables, in the encoding of x and y respectively, are not orthogonal.

We show how to convert *any* pair encoding that satisfies Sym-Prop into a *fully* secure encryption scheme whose security is based on a *fixed* q-type assumption that we call q-ratio. We use the generic transformation from Agrawal and Chase [AC16], henceforth called Gen-Trans, for this purpose. Gen-Trans takes an encoding scheme satisfying a certain information-theoretic property and produces an encryption scheme in dual system groups [CW14a], which can then be instantiated in composite-order groups under subgroup decision assumptions or prime-order groups under the k-linear assumption.

We show that the security of Gen-Trans can also be argued when the pair encoding satisfies a very different security property, the symbolic property. The main novelty in our proof, and the crucial difference from AC16, is in how the form of master secret key is changed: while AC16 uses an information-theoretic property, we use Sym-Prop in conjunction with a new assumption called $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ on dual system groups. [3] At a very high level, the terms that cannot be generated from $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ are exactly the ones that go to zero due to Sym-Prop. Thus we are able to embed $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ successfully into the reduction. Interestingly, however, as we will discuss below, Sym-Prop is not just an artifact of our proof strategy but seems to be inherently linked to the fundamental security of the resulting predicate encryption schemes.

An added advantage of borrowing AC16's transformation is that when a pair encoding is *used* in a way that can be shown to be information-theoretically secure, then the encryption scheme obtained through Gen-Trans is fully secure under a standard assumption. We show a useful application of this feature below.

We also show that the q-ratio assumption is in fact implied by several other q-type assumptions used to construct ABE schemes, in particular those used in the Lewko-Waters ABE [LW12] and Attrapadung's fully secure predicate encryption for regular languages [Att14a]. This assumption is also simpler to describe than either [LW12] or [Att14a] and we believe that this approach better captures the intuition for why these schemes are secure.

Analysis of pair encodings. We show that Sym-Prop holds for several pair encoding schemes, both new and old: multi-use CP-ABE, short ciphertext CP-ABE, large universe KP-ABE, short ciphertext KP-ABE, and predicate encryption for regular languages.

First, we present a new pair encoding $\Pi_{\mathsf{re\text{-}use}}$ for CP-ABE that allows an attribute to be used *any* number of times in a policy. An interesting feature of $\Pi_{\mathsf{re\text{-}use}}$ is that if no attribute is used more than once, then it collapses to the one-use scheme of [Att14a], which is information-theoretically secure. So if we get an encryption scheme ES when Gen-Trans is applied on $\Pi_{\mathsf{re\text{-}use}}$, then ES is fully secure under a *standard* assumption as long as it is used to encrypt policies where attributes are not repeated. If a policy with multiple use of attributes needs to be encrypted, then ES still fully hides the payload but under a q-type assumption. As far as we know, no multi-use scheme with this feature was known before. For instance, the Lewko-Waters' scheme [LW12] uses an assumption whose size scales with that of the access policy in the challenge ciphertext. So even if no attribute is used more than once, security still relies on a q-type assumption. [4]

For short ciphertext CP-ABE, we show that the pair encoding of Agrawal and Chase [AC16] satisfies Sym-Prop. This means that the encryption scheme that comes out after applying Gen-Trans is fully secure, not just selectively secure as they proved it (since we use the same transformation as

[3] $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ is very similar to q-ratio. We show that Chen and Wee's instantiations of dual system groups satisfy $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ if the underlying bilinear maps satisfy q-ratio.

[4] There are other ABE schemes that get much more than attribute re-use, like large universe or short keys, based on q-type assumptions [Att14a], but proving them secure under a standard assumption when re-use does not happen would be even more difficult.

them), under a q-type assumption. Note that it was not known earlier whether there exists a fully-secure CP-ABE scheme with constant-size ciphertexts under any kind of assumption on bi-linear maps. In fact, we can *generically* build an encryption scheme with constant-size ciphertexts for any predicate P from *any* pair encoding for P that satisfies Sym-Prop as discussed in more detail below.

The last three encodings we analyze are borrowed from the work of Attrapadung [Att14a] with slight simplification. Previously, we only knew how to analyze them using the much more complex computational security property in [Att14a]. Our analysis of these schemes is considerably simpler: for comparison, the proof of computational security for the regular languages pair encoding required 9 full pages, while our proof of symbolic security only takes 2.5 llncs pages. Our proofs can be seen as extracting, abstracting and somewhat simplifying the key ideas behind Attrapadung's security analysis, so that they can be very easily verified, and more easily applied to future schemes.

Symbolic property inherent in a secure scheme. While there are several security properties for encoding schemes that allow one to check if they can be used to build some type of encryption scheme, is there a property that an encoding scheme should *not* satisfy? A natural one that comes to mind is that correctness holds for an x and y that make a predicate *false*. In other words, there exists a way to combine the polynomials in the encoding to recover the blinding factor for the message even when the predicate is false. We call a pair encoding scheme that satisfies this property *trivially broken*.

Building an encryption scheme from a pair encoding scheme seems to require at least that the pair encoding *not* be trivially broken, but there is no general result that shows some type of security for a scheme that only provides such a minimal guarantee. In Section 4, we give the first result of this kind: *Any pair encoding scheme that is not trivially broken satisfies our symbolic property.*

This result has several interesting broad implications. Suppose we have an encoding Π that we do not know to be secure. We apply Gen-Trans on it to get an encryption scheme ES. For this scheme to not be completely broken, there should not be a way to trivially combine some ciphertext and key to recover the message when the predicate is false. Now an interesting fact about our generic transformation Gen-Trans is that it preserves the structure of pair encodings, so that if there is way to combine the polynomials to recover the blinding factor, then the ciphertext and key coming out of Gen-Trans can be combined to recover the message. Therefore, if ES is not completely broken, Π is not broken either. This further implies that Π satisfies Sym-Prop and ES is fully secure under q-ratio. Thus we arrive at a very interesting conclusion: *Either* ES *is broken in an obvious way or it is fully secure under* q-ratio. Hence, Sym-Prop seems to be inherently linked to the fundamental security of encryption schemes, and is not just an artifact of our proof strategy.

We can take this line of argument even further. Suppose there is a generic transformation that preserves the structure of pair encodings in the sense described above. And suppose that when an encoding scheme satisfying a certain property X is given as input, it generates an encryption scheme that is not obviously broken, for example a *selectively* secure scheme. Then every encoding that satisfies X will also satisfy our symbolic property, and hence will lead to a *fully* secure encryption scheme through Gen-Trans! In this paper, we do not formalize the exact requirements a generic transformation should satisfy for such a general result to hold, leaving it as an interesting exercise for future work.

We conclude with an alternate way of proving symbolic security in case finding a mapping from an encoding's variables to matrices/vectors seems difficult: show that for all x and y for which the predicate is false, the blinding factor cannot be recovered from the encoding's polynomials.

New generic conversions. Thanks to the simplicity of our new symbolic property, we are able to show several useful transformations of pair encodings that preserve security. Specifically,

1. *Dual conversion.* Any secure pair encoding for a predicate can be transformed into a secure encoding scheme for the dual predicate (where the role of key and ciphertext are switched).

2. *Compact ciphertexts.* Any secure pair encoding can be converted into one that has a constant number of variables and polynomials in the ciphertext encoding. Thus, after applying Gen-Trans to the latter encoding, one gets encryption schemes with constant-size ciphertexts.

3. *Compact keys.* Analogous to above, any secure pair encoding can be converted into one that has a constant number of variables and polynomials in the key encoding, leading to encryption schemes with constant-size keys. [5]

This demonstrates the power and versatility of the new symbolic property. In contrast, only the first type of transformation is known for the security properties of Attrapadung [Att14a, AY15], and none is known for Wee [Wee14] or Chen et al. [CGW15].

More new schemes. Apart from the new scheme for unbounded attribute-reuse and showing that the constant-size ciphertext CP-ABE of [AC16] is fully secure, our generic conversions for pair encodings help us arrive at schemes that were not known before:

- As mentioned before, we show that the regular language pair encoding from [Att14a] satisfies our symbolic property. Here keys are associated with regular languages, expressed as deterministic finite automata (DFA), and ciphertexts are associated with strings of any length from an alphabet set. One can first apply the dual conversion transformation to get an encoding scheme where ciphertexts and keys are associated with DFAs and strings, respectively. Then applying our compact ciphertext transformation to this encoding, and using the resulting pair encoding in Gen-Trans, one gets an encryption scheme for regular languages with constant sized ciphertexts (but with an upper bound on the size of DFAs).

- Similarly, applying our compact ciphertext/key transformation to Attrapadung's pair encodings for doubly spatial encryption (DSE) yields new encoding schemes, that then lead to encryption schemes with constant size ciphertext and keys, respectively. The only previous work on short ciphertext DSE [AHY15] relied on a more complex series of transformations in which one type of predicate family (e.g. CP-ABE) is embedded inside another (e.g. DSE), and resulted in more expensive encodings.

1.2 Overview of Symbolic Security

This section provides a high-level *informal* treatment of pair encodings and the symbolic property with the goal of building some intuition about these concepts. Please refer to Section 3 for a formal presentation.

Pair encodings. The pair encoding framework focuses on the exponent space of an encryption scheme. Suppose there is a predicate P that takes two inputs x and y. We want to encode x into a ciphertext and y into a key. An encryption scheme for P generally has terms like $g^{b_1}, g^{b_2}, \ldots$ and a special one of the form $e(g,g)^{\alpha}$ in the public parameters (b_1, b_2, … and α are chosen randomly).

[5] This transformation and the one above requires some bound on the number of variables and polynomials in the respective encoding.

α plays the role of the master secret key. To encrypt a message m along with attribute x, some random numbers $s_0, s_1, s_2, \ldots$ are chosen and new terms are created by raising g, or some *common* term like g^{b_j}, to some s_i, and then taking a linear combination of these terms, where the terms and combination used depend on x. So, if we look at the exponent of any group element output by the encryption algorithm, it is usually a polynomial of the form $s_1 + \lambda_1 s_2 b_3 + \ldots$ where λ_1 is a constant that depends on x. Finally, m is hidden inside the ciphertext by blinding it with a re-randomization of $e(g,g)^{\alpha}$, say $e(g,g)^{\alpha s_0}$.

Similarly, the exponents of group elements in any key are of the form $r_1 + \mu r_2 b_1 + \ldots$, where $r_1, r_2, \ldots$ is fresh randomness chosen for this key. We could also have expressions that contain α because key generation involves the master secret key. Thus there are three different types of variables involved in a pair encoding: the common variables $b_1, b_2, \ldots$, the ciphertext encoding variables $s_0, s_1, s_2, \ldots$, and the key encoding variables $\alpha, r_1, r_2, \ldots$.

Overall, it can be seen that if we focus on the exponent space of an encryption scheme, we need to deal with polynomials of a special form only. If $P(x, y) = 1$, then it should be possible to combine the ciphertext and key polynomials so that αs_0 can be recovered, and then used to unblind the message. The pair encoding framework just abstracts out such similarities between predicate encryption schemes in a formal way.

Security properties and transformation. Many security properties have been proposed in the literature for pair encodings, and a more restricted structure called predicate encodings [Wee14, Att14a, CGW15, AC16]. The main contribution of these papers is to give a *generic* transformation from *any* pair encoding that satisfies their respective property into a fully secure predicate encryption scheme in composite or prime order groups (or a higher level abstraction called dual-system groups [CW14a]). Proving that a pair encoding scheme satisfies a certain property is *significantly* easier, especially if the property is information-theoretic, than directly proving security of an encryption scheme. This is not surprising because there are no bi-linear maps, hardness assumptions, or sophisticated dual-encryption techniques involved in this process. Furthermore, verifying security of any number of encryption schemes designed through the pair encoding framework reduces to checking that the respective pair encodings are secure—a much easier task—and that the generic transformation is correct—a one-time effort. Needless to say, this saves a huge amount of work.

A concrete example: Unbounded attribute re-use. Suppose we want to design an ABE scheme that puts *no* restriction on the number of times an attribute can be used in an access policy. We know that a linear secret sharing scheme is the standard way to present a policy. It consists of a matrix $\mathbf{A}$ of size $m \times k$ and a mapping π from its rows to the universe of attributes. A value γ can be secret-shared through $\mathbf{A}$ by creating m shares, one for each row. If a user has a set of attributes S, then she gets shares for all the rows that map to some attribute in S through π. If S satisfies $(\mathbf{A}, \pi)$, then those shares can be combined to recover γ; otherwise, γ is information-theoretically hidden. In nearly all fully secure ABE schemes, the mapping π is assumed to be injective or one-to-one (this is called the one-use restriction), but we want to build an ABE scheme that supports any π whatsoever. In particular, the size of public parameters should not affect how many times an attribute can be used in a policy. (Any such scheme will likely rely on a q-type assumption [LW12].[6])

[6] In a recent work, Kowalczyk and Lewko [KL15] proposed a new technique to boost the entropy of a small set of (unpublished) semi-functional parameters. Using this idea, they propose a new KP-ABE scheme where the number of group elements in the public parameters grows only logarithmically in the bound on the number of attribute-uses in a policy, but note that the number of times an attribute can be reused is still affected. Furthermore, the size of ciphertexts scales with the maximum number of times an attribute can be re-used.

For a row i of $\mathbf{A}$, suppose $\rho(i)$ denotes which occurrence of $\pi(i)$ this is. (If an attribute y is attached to the second and fifth rows, then $\rho(2) = 1$ and $\rho(5) = 2$.) We now present a new pair encoding $\Pi_{\mathsf{re\text{-}use}}$ for unbounded re-use by adapting the one-use scheme of [Att14a]. (Some minor elements of the encoding have been suppressed for simplicity; see Appendix B.1 for a full description.)

$$\begin{aligned} \mathsf{EncCt}((\mathbf{A},\pi)) &\to s_0, s_1, \ldots, s_d, \quad \{\mathbf{a}_i(s_0 b', \hat{s}_2, \ldots, \hat{s}_k)^{\top} + s_{\rho(i)} b_{\pi(i)}\}_{i=1,\ldots,m} \\ \mathsf{EncKey}(S) &\to r, \quad \alpha + r b', \quad \{r b_y\}_{y \in S} \end{aligned}$$

Here $\mathbf{a}_i$ is the ith row of $\mathbf{A}$ and d is the maximum number of times any attribute appears in it. A nice feature of $\Pi_{\mathsf{re\text{-}use}}$ is that if no attribute is used more than once (i.e. $d = 1$), then the scheme collapses to that of [Att14a], and one can show that α is information-theoretically hidden, or that $\Pi_{\mathsf{re\text{-}use}}$ is *perfectly* secure.

If attributes are used multiple times, so that the ciphertext encoding has several variables $s_1, \ldots, s_d$, then α might be revealed to an unbounded adversary. Thus we need to find out if $\Pi_{\mathsf{re\text{-}use}}$ satisfies a different type of property for which a generic transformation is known. One possibility is the computational *double selective master-key hiding* property due to Attrapadung, but then the advantages of an abstraction like pair encoding are more or less lost: we will have to work at the level of bi-linear maps instead of simple polynomials, and find a suitable q-type assumption(s) under which the property can be shown to hold.

The symbolic property. Our new symbolic property (Sym-Prop) can be very useful in such cases. It provides a new, clean way of reasoning about security of pair encodings: instead of arguing that one distribution is indistinguishable from another, whether information-theoretically or computationally, one needs to discover a mapping from the variables involved in an encoding to matrices and vectors, such that when the latter is substituted for the former in any ciphertext/key encoding polynomial, the zero vector is obtained. Indeed, one needs to invest some effort in order to find the right matrices and vectors that will make the polynomials go to zero, but once such a discovery is made, verifying the property is just a matter of doing some simple linear algebra.

Recall that a pair encoding scheme for a predicate P that takes two inputs x and y, consists of three different types of variables: common variables $b_1, b_2, \ldots$, ciphertext encoding variables $s_0, s_1, s_2, \ldots$, and key encoding variables $\alpha, r_1, r_2, \ldots$. Sym-Prop is defined w.r.t. three (deterministic) algorithms, EncB, EncS and EncR. Among them, EncB generates matrices for the common variables; EncS and EncR generate vectors for ciphertext encoding and key encoding variables, respectively. The inputs to these three algorithms depend on what type of symbolic property we want to prove. For the selective version, the three algorithms get x as input, while EncR also gets y; and for the co-selective version, they all get y as input, while EncS also gets x. This is in line with the selective and co-selective security notions for encryption schemes. In the former, all key queries come after the challenge ciphertext, while in the latter, they come beforehand. A pair encoding scheme satisfies Sym-Prop if it satisfies both the selective and co-selective variants.

The trivial case where all the matrices and vectors output by the three algorithms are simply zero is ruled out because we also require that the vectors corresponding to two special variables, s_0 in the encoding of x and α in the encoding of y, are not orthogonal.

Proving the symbolic property for $\Pi_{\mathsf{re\text{-}use}}$. To prove Sym-Prop for the multi-use encoding scheme $\Pi_{\mathsf{re\text{-}use}}$ defined above, we need to define the outputs of the three algorithms EncB, EncS and EncR (in other words, a mapping from the variables in $\Pi_{\mathsf{re\text{-}use}}$ to vectors and matrices) in both the selective and co-selective settings. Towards this, we make use of a simple combinatorial fact that is often used in arguing security of ABE schemes. If a set of attributes S does not satisfy an access policy $(\mathbf{A}, \pi)$, then there exists a vector $\mathbf{w} = (w_1, \ldots, w_k)$ s.t. $w_1 = 1$ and $\mathbf{a}_i$ is orthogonal to $\mathbf{w}$ for all i such that $\pi(i) \in S$. Note that $\mathbf{w}$ can be computed only by an algorithm that knows both $(\mathbf{A}, \pi)$ and S.

We also need some simple notation to describe the mapping. Let $\mathbf{E}_{i,j}$ be an $k \times d$ matrix with 1 at the (i, j)-th position and 0 everywhere else. Also, let $\mathbf{e}_j$ be the jth d-length unit vector and $\overline{\mathbf{e}}_i$ be the ith k-length unit vector. Here is the mapping for the selective version:

$$b_y : -\sum_{\ell=1}^{d} \sum_{j=1}^{k} a_{\sigma(y,\ell),j} \mathbf{E}_{j,\ell}, \qquad b' : \mathbf{E}_{1,1},$$
$$s_0 : \mathbf{e}_1, \qquad s_\ell : \mathbf{e}_\ell, \qquad \hat{s}_j : \overline{\mathbf{e}}_j, \qquad \alpha : \mathbf{e}_1, \qquad r : -\sum_{j=1}^{k} w_j \overline{\mathbf{e}}_j,$$

where $\sigma(y, \ell)$ is the index of the row in $\mathbf{A}$ which has the ℓ-th occurrence of y. Further, if $\mathbf{E}_{i,j}$, $\mathbf{e}_j$ and $\overline{\mathbf{e}}_i$ carry the same meaning as above, except that their dimensions are $1 \times T$, T and 1 respectively[7], then the mapping for the co-selective version is:

$$b_y : \mathbf{0} \text{ for } y \in S \text{ and } -\mathbf{E}_{1,y} \text{ otherwise}, \qquad b' : \mathbf{E}_{1,1},$$
$$s_0 : w_1 \mathbf{e}_1, \qquad s_\ell : \sum_{i:\rho(i)=\ell} \mathbf{a}_i \mathbf{w}^\top \mathbf{e}_{\pi(i)}, \qquad \hat{s}_j : w_j \overline{\mathbf{e}}_1, \qquad \alpha : \mathbf{e}_1, \qquad r : -\overline{\mathbf{e}}_1.$$

We encourage the reader to verify that the polynomials in $\Pi_{\mathsf{re\text{-}use}}$ (except the simples ones $s_0, s_1, \ldots, s_d, r$) go to zero when the two mappings described above are applied. (Vectors output by EncS (resp. EncR) are multiplied to the right (resp. left) of matrices output by EncB.) All it takes are simple observations like $\mathbf{E}_{i,j} \cdot \mathbf{e}_{j'}^\top$ gives a non-zero vector if and only if $j = j'$, and that $\mathbf{w}$ is orthogonal to every row in $\mathbf{A}$ that maps to an attribute in S. (See Appendix B.1 for a formal proof.) One can consider the two mappings to be a short *certificate* of the security of $\Pi_{\mathsf{re\text{-}use}}$.

How to find a mapping? Indeed, as pointed out earlier, finding an appropriate mapping is not a trivial task. Nevertheless, Sym-Prop is still the *right* property for arguing security of pair encodings for the following reasons:

- If finding the right mapping is difficult for Sym-Prop, then finding a proof for the computational property of Attrapadung [Att14a] is several times more difficult. A typical proof of the symbolic property is 1-2 pages while computational property proofs could go up to 10 pages (see the encoding for regular languages, for instance). A central issue with computational properties is finding an appropriate q-type assumption under which it holds, which may be very difficult for a complex predicate. Our approach can be seen as extracting out the *real* challenging part of designing Attrapadung's computational proofs.

- Verification of Sym-Prop involves doing simple linear algebra, arguably a much simpler task than checking indistinguishability of distributions, and certainly a much simpler task than verifying a long computational reduction.

- The *certificate* for the symbolic security of $\Pi_{\mathsf{re\text{-}use}}$ bears many similarities with those of other encodings that we will describe later in the paper. Thus proving Sym-Prop for a new encoding scheme is not as difficult as it might seem at first. Furthermore, modifying a short proof of the symbolic property is much easier than a long proof of a computational property.

- Recall our result that if an encoding scheme is not trivially broken then it satisfies Sym-Prop. This gives an alternate way of showing that Sym-Prop holds, by proving that the scheme is not broken.

[7] It is assumed that the attributes are in the set $\{1, \ldots, T\}$.

1.3 Outline of The Paper

In Section 2 we define relevant notation and review the standard definition of predicate encryption. In Section 3 we define pair encoding schemes and our new symbolic property formally. Section 5 first reviews the notion of dual system groups, then shows how to build encryption schemes from any pair encoding by using them. This conversion is a two-step process: first we *augment* an encoding so that it satisfies a few extra properties (Section 5.1); next we apply the transformation from Agrawal and Chase [AC16] (Section 5.4). A proof of security of the resulting encryption scheme is provided in Section 7.

Section 6 gives generic transformations that can be used to reduce the number of variables and/or polynomials in an encoding, which can then be used to get encryption schemes with constant-size ciphertexts/keys. We also provide a transformation from any encoding for a predicate to an encoding for the dual predicate.

In Appendix B, we show how symbolic property can substantially simplifying the analysis of encoding schemes for complex predicates by giving several examples. Finally, we discuss the new schemes that we get through our various transformations in Appendix E.

2 Preliminaries

We use λ to denote the security parameter. A negligible function is denoted by negl. We use bold letters to denote matrices and vectors, with the former in uppercase and the latter in lowercase. The operator $\cdot$ applied to two vectors computes their entry-wise product and $\langle,\rangle$ gives the inner-product. For a vector $\mathbf{u}$, we use u_i to denote its ith element, and for a matrix $\mathbf{M}$, $M_{i,j}$ denotes the element in the ith row and jth column. When we write $g^{\mathbf{u}}$ for a vector $u = (u_1, \ldots, u_n)$, we mean the vector $(g^{u_1}, \ldots, g^{u_n})$. $g^{\mathbf{M}}$ for a matrix $\mathbf{M}$ should be interpreted in a similar way. The default interpretation of a vector should be as a row vector.

For two matrices $\mathbf{U}$ and $\mathbf{V}$ of dimension $n \times m_1$ and $n \times m_2$ respectively, let $\mathbf{U} \circ \mathbf{V}$ denote the column-wise *join* of $\mathbf{U}$ and $\mathbf{V}$ of dimension $n \times (m_1 + m_2)$, i.e., $\mathbf{U} \circ \mathbf{V}$ has the matrix $\mathbf{U}$ as the first m_1 columns and $\mathbf{V}$ as the remaining m_2 columns. We also refer to this operation as *appending* $\mathbf{V}$ to $\mathbf{U}$. (The notation easily extends to vectors because we represent them as row matrices.) If we want to join matrices row-wise instead, we could take their transpose, apply a column-wise join, and then take the transpose of the resultant matrix.

We use $x \leftarrow_R S$, for a set S, to denote that x has been drawn uniformly at random from it. The set of integers $a, a+1, \ldots, b$ is compactly represented as $[a, b]$. If $a = 1$, then we just use $[b]$, and if $a = 0$, then $[b]^+$.

Let $\mathbb{Z}_N$ denote the set of integers $\{0, 1, 2, \ldots, N\}$. Let $\mathcal{G}_N(m)$ denote the set of all vectors of length m with every element in $\mathbb{Z}_N$. Similarly, let $\mathcal{G}_N(m_1, m_2)$ denote the set of all matrices of size $m_1 \times m_2$ that have all the elements in $\mathbb{Z}_N$.

Indistinguishability. A function $f(\cdot)$ is *negligible* if $f(n) < n^{-c}$ for every constant $c > 0$ and sufficiently large n. The *statistical distance* between two discrete probability distributions X and Y, denoted by $\Delta(X, Y)$, is the maximum value of $|\Pr[A(X) = 1] - \Pr[A(Y) = 1]$ over all functions A. Two distribution ensembles $\{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $\{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are *perfectly indistinguishable*, denoted by $\{X_\lambda\}_{\lambda \in \mathbb{N}} \equiv \{Y_\lambda\}_{\lambda \in \mathbb{N}}$, if $\Delta(X_\lambda, Y_\lambda)$ is equal to 0 for any λ. They are *statistically indistinguishable* if the statistical distance is negligible as a function of λ. We use $\cong$ to denote statistical indistinguishability. Finally, if for every (non-uniform) probabilistic polynomial time (PPT) algorithm $\mathcal{A}$, the distinguishing advantage $|\Pr[\mathcal{A}(1^\lambda, X_\lambda) = 1] - \Pr[\mathcal{A}(1^\lambda, Y_\lambda) = 1]$ is negligible in λ, then we say that X_λ, Y_λ are computationally indistinguishable, and denote it by $X_\lambda \approx Y_\lambda$ for simplicity.

Bilinear Pairings: We use the standard definition of pairing friendly groups from literature. A mapping e from a pair of groups $(\mathcal{G}, \mathcal{H})$ to a target group $\mathcal{G}_T$ is bilinear if there is linearity in both the first and second inputs, i.e. $e(g^a, h^b) = e(g, h)^{ab}$ for every $g \in \mathcal{G}, h \in \mathcal{H}$ and $a, b \in \mathbb{Z}$. We require e to be non-degenerate and efficiently computable. The identity element of a group G is denoted by 1_G.

Let $\mathsf{GroupGen}$ be an algorithm that on input the security parameter λ outputs $(N, \mathcal{G}, \mathcal{H}, \mathcal{G}_T, g, h, e)$ where $N = \Theta(\lambda)$; $\mathcal{G}$, $\mathcal{H}$ and $\mathcal{G}_T$ are (multiplicative) cyclic groups of order N; g, h are generators of $\mathcal{G}$, $\mathcal{H}$, respectively; and $e : \mathcal{G} \times \mathcal{H} \to \mathcal{G}_T$ is a bilinear map. In this paper our focus will be on prime-order groups because they perform much better in practice.

Predicate family. We borrow the notation of predicate family from Attrapadung [Att14a]. It is given by $P = \{P_\kappa\}_{\kappa \in \mathbb{N}^c}$ for some constant c, where P_κ maps an $x \in \mathcal{X}_\kappa$ and a $y \in \mathcal{Y}_\kappa$ to either 0 or 1. The first entry of κ is a number $N \in \mathbb{N}$ that is supposed to specify the size of a domain; rest of the entries are collectively referred to as par, i.e. $\kappa = (N, \mathsf{par})$.

2.1 Predicate Encryption

An encryption scheme for a predicate family $P = \{P_\kappa\}_{\kappa \in \mathbb{N}^c}$ over a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ consists of a tuple of four PPT algorithms $(\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{KeyGen}, \mathsf{Decrypt})$ that satisfy a correctness condition. These algorithms behave as follows.

- $\mathsf{Setup}(1^\lambda, \mathsf{par})$. On input 1^λ and par, Setup outputs a master public key $\textsc{mpk}$ and a master secret key $\textsc{msk}$. The output of Setup is assumed to also define a natural number N, and κ is set to (N, par).

- $\mathsf{Encrypt}(\textsc{mpk}, x, m)$. On input $\textsc{mpk}$, $x \in \mathcal{X}_\kappa$ and $m \in \mathcal{M}_\lambda$, $\mathsf{Encrypt}$ outputs a ciphertext $\textsc{ct}$.

- $\mathsf{KeyGen}(\textsc{msk}, y)$. On input $\textsc{msk}$ and $y \in \mathcal{Y}_\kappa$, KeyGen outputs a secret key $\textsc{sk}$.

- $\mathsf{Decrypt}(\textsc{mpk}, \textsc{sk}, \textsc{ct})$. On input $\textsc{mpk}$, a secret key $\textsc{sk}$ and a ciphertext $\textsc{ct}$, $\mathsf{Decrypt}$ outputs a message $m' \in \mathcal{M}_\lambda$ or $\perp$.

Correctness: For all par, $m \in \mathcal{M}_\lambda$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $P_\kappa(x, y) = 1$,

$$\Pr[(\textsc{mpk}, \textsc{msk}) \leftarrow \mathsf{Setup}(1^\lambda); \\ \mathsf{Decrypt}(\textsc{mpk}, \mathsf{KeyGen}(\textsc{msk}, y), \mathsf{Encrypt}(\textsc{mpk}, x)) \neq P_\kappa(x, y)] \leq \mathsf{negl}(\lambda),$$

where the probability is over the random coin tosses of Setup, $\mathsf{Encrypt}$ and KeyGen ($\mathsf{Decrypt}$ can be assumed to be deterministic without loss of generality).

Security: Consider the following game $\mathsf{IND\text{-}CPA}^b_{\mathcal{A}}(\lambda, \mathsf{par})$ between a challenger Chal and an adversary $\mathcal{A}$ for $b \in \{0, 1\}$ when both are given inputs 1^λ and par:

1. *Setup Phase*: Chal runs $\mathsf{Setup}(1^\lambda, \mathsf{par})$ to obtain $\textsc{mpk}$ and $\textsc{msk}$. It gives $\textsc{mpk}$ to $\mathcal{A}$.

2. *Query Phase*: $\mathcal{A}$ requests a key by sending $y \in \mathcal{Y}_\kappa$ to Chal, and obtains $\textsc{sk} \leftarrow \mathsf{KeyGen}(\textsc{msk}, y)$ in response. This step can be repeated any number of times.

3. *Challenge Phase*: $\mathcal{A}$ sends two messages $m_0, m_1 \in \mathcal{M}_\lambda$ and an $x^\star \in \mathcal{X}_\kappa$ to Chal, and gets $\textsc{ct} \leftarrow \mathsf{Encrypt}(\textsc{mpk}, x, m_b)$ as the challenge ciphertext.

4. *Query Phase*: This is identical to step 2.

5. *Output.* $\mathcal{A}$ outputs a bit.

The output of the experiment is the bit that $\mathcal{A}$ outputs at the end. It is required that for all y queried in steps 2 and 4, $P_\kappa(x^\star, y) = 0$.

Definition 2.1. An encryption scheme is *adaptively* or *fully* secure if for all par and PPT adversary $\mathcal{A}$,

$$|\Pr[\mathsf{IND\text{-}CPA}^0_{\mathcal{A}}(\lambda, \mathsf{par}) = 1] - \Pr[\mathsf{IND\text{-}CPA}^1_{\mathcal{A}}(\lambda, \mathsf{par}) = 1]| \leq \mathsf{negl}(\lambda), \tag{1}$$

where the probabilities are taken over the coin tosses of $\mathcal{A}$ and Chal. It is *semi-adaptively* secure if (1) is satisfied with respect to a modified version of IND-CPA where the second step is omitted [CW14b, Wat15]. Further, it is *co-selectively* secure if (1) holds when the fourth step is removed from the IND-CPA game [AL10].

3 Pair Encoding Schemes

The notion of pair encoding schemes (PES) was introduced by Attrapadung [Att14a], and later refined independently by Agrawal and Chase [AC16] and Attrapadung [Att16] himself in an identical way. As observed in the latter works, *all* pair encodings proposed originally in [Att14a] satisfy the additional constraints in the refined versions.

We present here a more structured definition of pair encoding schemes so that the reader can easily see the different components involved. In Appendix A we describe the original formulation as well, and argue why our definition does not lose any generality.

3.1 Definition

A PES for a predicate family $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ indexed by $\kappa = (N, \mathsf{par})$, where par specifies some parameters, is given by four *deterministic* polynomial-time algorithms as described below.

- $\mathsf{Param}(\mathsf{par}) \to n$. When given par as input, Param outputs $n \in \mathbb{N}$ that specifies the number of *common* variables, which we denote by $\mathbf{b} := (b_1, \ldots, b_n)$.

- $\mathsf{EncCt}(x, N) \to (w_1, w_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}))$. On input $N \in \mathbb{N}$ and $x \in \mathcal{X}_{(N,\mathsf{par})}$, EncCt outputs a vector of polynomials $\mathbf{c} = (c_1, \ldots, c_{w_3})$ in *non-lone* variables $\mathbf{s} = (s_0, s_1, \ldots, s_{w_1})$ and *lone* variables $\hat{\mathbf{s}} = (\hat{s}_1, \ldots, \hat{s}_{w_2})$. (The variables $\hat{s}_1, \ldots, \hat{s}_{w_2}$ never appear in the form $\hat{s}_z b_j$, and are hence called lone.) For $\ell \in [w_3]$, where $\eta_{\ell,z}, \eta_{\ell,i,j} \in \mathbb{Z}_N$, the ℓth polynomial is given by

$$\sum_{z \in [w_2]} \eta_{\ell,z} \hat{s}_z \quad + \sum_{\substack{i \in [w_1]^+, \\ j \in [n]}} \eta_{\ell,i,j} s_i b_j.$$

- $\mathsf{EncKey}(y, N) \to (m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}))$. On input $N \in \mathbb{N}$ and $y \in \mathcal{Y}_{(N,\mathsf{par})}$, EncKey outputs a vector of polynomials $\mathbf{k} = (k_1, \ldots, k_{m_3})$ in non-lone variables $\mathbf{r} = (r_1, \ldots, r_{m_1})$ and lone variables $\hat{\mathbf{r}} = (\alpha, \hat{r}_1, \ldots, \hat{r}_{m_2})$. For $t \in [m_3]$, where $\phi_t, \phi_{t,z'}, \phi_{t,i',j} \in \mathbb{Z}_N$ the tth polynomial is given by

$$\phi_t \alpha \quad + \sum_{z' \in [m_2]} \phi_{t,z'} \hat{r}_{z'} \quad + \sum_{\substack{i' \in [m_1], \\ j \in [n]}} \phi_{t,i',j} r_{i'} b_j.$$

- $\mathsf{Pair}(x, y, N) \to (\mathbf{E}, \overline{\mathbf{E}})$. On input N, and both x and y, Pair outputs two matrices $\mathbf{E}$ and $\overline{\mathbf{E}}$ of size $(w_1+1) \times m_3$ and $w_3 \times m_1$, respectively.

Observe that the output of EncKey is analogous to that of EncCt, except in how the special variables α and s_0 are treated in the respective case. While α is lone variable, i.e. it never appears in conjunction with a common variable, s_0 is not. See Appendix B for several concrete examples of pair encodings and the different types of variables involved.

Correctness. A PES is correct if for every $\kappa = (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $P_\kappa(x, y) = 1$, the following holds symbolically

$$\mathbf{s}\mathbf{E}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top \quad = \sum_{\substack{i \in [w_1]^+, \\ t \in [m_3]}} s_i E_{i,t} k_t \quad + \sum_{\substack{\ell \in [w_3], \\ i' \in [m_1]}} c_\ell \overline{E}_{\ell,i'} r_{i'} \quad = \quad \alpha s_0.$$

The matrix $\mathbf{E}$ takes a linear combination of the products of non-lone variables output by EncCt and polynomials output by EncKey. (Its rows are numbered from 0 to w_1.) Analogously, $\overline{\mathbf{E}}$ takes a linear combination of the products of polynomials output by EncCt and non-lone variables output by EncKey. Below we use ct-enc and key-enc as a shorthand for polynomials and variables output by EncCt (ciphertext-encoding) and EncKey (key-encoding), respectively.

3.2 Symbolic Property

We introduce a new symbolic property for pair encoding schemes that significantly simplifies their analysis for even complex predicates. We get the best of two worlds: not only is our symbolic property very clean to describe (like information-theoretic properties), it can also capture all the predicates that have been previously captured by any computational property. Further, the property does not involve dealing with any kind of distribution.

We now formally define the property. We use $a : b$ below to denote that a variable a is substituted by a matrix/vector b.

Definition 3.1 (Symbolic property)**.** *A pair encoding scheme* $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ *for a predicate family* $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ *satisfies* (d_1, d_2)-selective symbolic property[8] *for positive integers* d_1 *and* d_2 *if there exist three deterministic polynomial-time algorithms* EncB, EncS, EncR *such that for all* $\kappa = (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$, $y \in \mathcal{Y}_\kappa$ *with* $P_\kappa(x, y) = 0$,

- $\mathsf{EncB}(x) \to \mathbf{B}_1, \ldots, \mathbf{B}_n \in \mathcal{G}_N(d_1, d_2)$*;*
- $\mathsf{EncS}(x) \to \mathbf{s}_0, \ldots, \mathbf{s}_{w_1} \in \mathcal{G}_N(d_2), \quad \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2} \in \mathcal{G}_N(d_1)$*;*
- $\mathsf{EncR}(x, y) \to \mathbf{r}_1, \ldots, \mathbf{r}_{m_1} \in \mathcal{G}_N(d_1), \quad \mathbf{a}, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2} \in \mathcal{G}_N(d_2)$*;*

such that $\langle \mathbf{s}_0, \mathbf{a} \rangle \neq 0$*, and if we substitute*

$$\hat{s}_z : \hat{\mathbf{s}}_z^\top \qquad s_i b_j : \mathbf{B}_j \mathbf{s}_i^\top \qquad \alpha : \mathbf{a} \qquad \hat{r}_{z'} : \hat{\mathbf{r}}_{z'} \qquad r_{i'} b_j : \mathbf{r}_{i'} \mathbf{B}_j$$

for $z \in [w_2]$, $i \in [w_1]^+$, $j \in [n]$, $z' \in [m_2]$ *and* $i' \in [m_1]$ *in all the polynomials output by* EncCt *and* EncKey *on input x and y, respectively, they evaluate to* $\mathbf{0}$.

Similarly we say a pair encoding scheme satisfies (d_1, d_2)-co-selective symbolic security property *if there exist* EncB, EncR, EncS *that satisfy the above properties but where* EncB *and* EncR *depend only on y, and* EncS *depends on both x and y. Finally, a scheme satisfies* (d_1, d_2)-symbolic property *if it satisfies both* (d_1', d_2')*-selective and* (d_1'', d_2'')*-co-selective properties for some* $d_1', d_1'' \leq d_1$ *and* $d_2', d_2'' \leq d_2$.

[8] d_1, d_2 could depend on κ but we leave this implicit for simplicity of presentation.

We use $\mathsf{Sym\text{-}Prop}$ as a shorthand for symbolic property. It is easy to see that if a scheme satisfies (d_1,d_2)-selective $\mathsf{Sym\text{-}Prop}$ then it also satisfies (d_1',d_2') for any $d_1' \geq d_1$ and $d_2' \geq d_2$. Just append $d_1' - d_1$ rows of zeroes and $d_2' - d_2$ columns of zeroes to the $\mathbf{B}_j$ matrices, $d_2' - d_2$ zeroes to the $\mathbf{s}_i$ vectors, $d_1' - d_1$ zeroes to the $\hat{\mathbf{s}}_z$ vectors, $d_1' - d_1$ zeroes to the $\mathbf{r}_{i'}$ vectors, and $d_2' - d_2$ zeroes to the $\hat{\mathbf{r}}_{z'}$ vectors. A similar claim can also be made about co-selective $\mathsf{Sym\text{-}Prop}$. Thus if a PES satisfies (d_1,d_2)-$\mathsf{Sym\text{-}Prop}$ then it also satisfies selective and co-selective properties with the same parameters, as well as (d_1',d_2')-$\mathsf{Sym\text{-}Prop}$ for any $d_1' \geq d_1$ and $d_2' \geq d_2$.

Lastly, if a PES Γ satisfies $\mathsf{Sym\text{-}Prop}$ for a predicate family P_κ, we say that Γ is *symbolically secure* for P_κ, or simply that Γ is symbolically secure if the predicate family is clear from context.

4 Obtaining Symbolic Security Generically

In this section, we prove an interesting and useful result. If a pair encoding scheme in not *trivially* broken in the sense that for any x, y that do not satisfy the predicate, there does not exist a way to directly recover αs_0 from the encoding polynomials (note that for correctness we require exactly this, but when the predicate is true), then the scheme satisfies the symbolic property.

Definition 4.1 (Trivially broken scheme)**.** *A pair encoding scheme* $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ *for a predicate family* $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ *is* trivially broken *if for a* $\kappa = (N, \mathrm{par})$, $x \in \mathcal{X}_\kappa$, $y \in \mathcal{Y}_\kappa$ *that satisfy* $P_\kappa(x,y) = 0$, *there exists a matrix* $\mathbf{E}$ *such that* $(\mathbf{s},\mathbf{c})\mathbf{E}(\mathbf{r},\mathbf{k})^\top = \alpha s_0$, *where* $\mathbf{c}$ *is the vector of polynomials output by* $\mathsf{EncCt}(x,N)$ *in variables* $\mathbf{s} = (s_0,\ldots)$, $\hat{\mathbf{s}}$, $\mathbf{b}$, *and* $\mathbf{k}$ *is the vector of polynomials output by* $\mathsf{EncKey}(y,N)$ *in variables* $\mathbf{r}$, $\hat{\mathbf{r}} = (\alpha,\ldots)$, $\mathbf{b}$.

Theorem 4.2. *If a pair encoding scheme is not trivially broken then it satisfies the symbolic property.*

Proof. If a scheme Γ is not trivially broken, then for all x and y for which the predicate evaluates to false, the $\mathsf{ct\text{-}enc}$ non-lone variables $\mathbf{s} = (s_0,\ldots,s_{w_1})$ and polynomials $\mathbf{c} = (c_1,\ldots,c_{w_3})$ cannot be paired with the $\mathsf{key\text{-}enc}$ non-lone variables $\mathbf{r} = (r_1,\ldots,r_{m_1})$ and polynomials $\mathbf{k} = (k_1,\ldots,k_{m_3})$ to recover αs_0. We know that the former have monomials of the form $s_0,\ldots,s_{w_1}$, $\hat{s}_1,\ldots,\hat{s}_{w_2}$, $s_0 b_1,\ldots,s_0 b_n$, $\ldots$, $s_{w_1} b_1,\ldots,s_{w_1} b_n$, so a total of $w_2 + (n+1)(w_1+1)$. Similarly, the total number of distinct monomials in the latter is $m_2 + 1 + (n+1)m_1$ (because α is a lone variable as opposed to s_0). Let us denote the two quantities above by var_c and var_k respectively.

Define a matrix Δ over $\mathbb{Z}_N$ with $(w_1 + w_3 + 1)(m_1 + m_3)$ rows and $\mathsf{var}_c\mathsf{var}_k$ columns. A row is associated with the product of a $\mathsf{ct\text{-}enc}$ non-lone variable or polynomial with a $\mathsf{key\text{-}enc}$ non-lone variable or polynomial. Each column represents a unique monomial that can be obtained by multiplying a $\mathsf{ct\text{-}enc}$ monomial with a $\mathsf{key\text{-}enc}$ monomial, with the first column representing αs_0. The (i,j)th entry in this matrix is the coefficient of the monomial associated with the jth column in the product polynomial attached with the ith row. Since Γ is not broken, we know that the rows in Δ cannot be linearly combined to get the vector $(1,0,\ldots,0)$.

Note that it is enough to work with any subset of rows because they cannot be combined to get $(1,0,\ldots,0)$ either. Thus, for the rest of the proof, we consider only those rows of Δ that multiply a $\mathsf{ct\text{-}enc}$ non-lone variable with a $\mathsf{key\text{-}enc}$ polynomial and vice versa (and only those columns which have monomials that can be obtained from multiplying such polynomials). Let n_1 denote the number of rows now.

Since rows in Δ cannot be linearly combined to get $(1,0,\ldots,0)$, the first column of Δ, say col, can be written as a linear combination of the other columns. Because if not, one can show that there exists a vector $\mathbf{v} = (v_1,\ldots,v_{n_1})$ that is orthogonal to all the columns except the first one[9]. We can then combine the rows of Δ using $v_1 / \langle \mathsf{col}, \mathbf{v}\rangle, \ldots, v_{n_1} / \langle \mathsf{col}, \mathbf{v}\rangle$ to get $(1,0,\ldots,0)$—a contradiction.

[9] The claim is similar to one made in the case of linear secret sharing schemes where we say that if a set of attributes

Let $\mathcal{Q}$ denote the set of monomials associated with the columns of Δ. These columns can be linearly combined to get the zero vector, without zeroing out col, which corresponds to αs_0. Let λ_q be the factor that multiplies the column associated with the monomial $q \in \mathcal{Q}$ in one such linear combination. Note that $\lambda_{\alpha s_0} \neq 0$.

Our first goal is to show that Γ satisfies the selective symbolic property. So we need to define matrices and vectors for various variables in the encoding such that all the polynomials evaluate to the zero vector. Towards this, pick any non-lone key-enc variable $r_{i'}$ for $i' \in [m_1]$ and consider the sub-matrix Δ' of Δ that consists of rows which are attached with the product of $r_{i'}$ with a ct-enc polynomial and columns which are associated with the product of $r_{i'}$ and a ct-enc monomial. (Note that it does not matter which non-lone key-enc variable we consider; the sub-matrix obtained in each case will be exactly the same.) Recall that a ct-enc polynomial c_ℓ is given by

$$\sum_{z \in [w_2]} \eta_{\ell,z} \hat{s}_z \quad + \sum_{i \in [w_1]^+, j \in [n]} \eta_{\ell,i,j} s_i b_j$$

for $\ell \in [w_3]$. So more formally, rows in Δ' are associated with $(c_\ell, r_{i'})$, and columns are associated with monomials $\hat{s}_z r_{i'}$, $s_i b_j r_{i'}$, where the range of i, j, z is as described above. For simplicity in the following, assume that the columns are ordered as $\hat{s}_1, \ldots, \hat{s}_{w_2}$, $s_0 b_1, \ldots, s_0, b_n, \ldots, s_{w_1} b_1, \ldots, s_{w_1} b_n$ and the rows are ordered as $(c_1, r_{i'}), \ldots, (c_{w_3}, r_{i'})$, so that the lth row of Δ' is $(\eta_{\ell,1}, \ldots, \eta_{\ell,w_2}, \eta_{\ell,0,1}, \ldots, \eta_{\ell,0,n}, \ldots, \eta_{\ell,w_1,1}, \ldots, \eta_{\ell,w_1,n})$.

Let $\mathcal{T}$ be the kernel of Δ', i.e. the set of all vectors $\mathbf{v}$ such that $\Delta' \mathbf{v} = \mathbf{0}$. Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{d_1}$ be a basis of $\mathcal{T}$ and write $\mathbf{v}_p$ as $(v_{p,1}, \ldots, v_{p,w_2}, v_{p,0,1}, \ldots, v_{p,0,n}, \ldots, v_{p,w_1,1}, \ldots, v_{p,w_1,n})$ for $p \in [d_1]$. (We discuss the special case of Δ''s kernel being empty later on.) Therefore, we have that for any $\ell \in [w_3]$ and $p \in [d_1]$,

$$\sum_z \eta_{\ell,z} v_{p,z} \quad + \quad \sum_{i,j} \eta_{\ell,i,j} v_{p,i,j} \tag{2}$$

is equal to 0. Let $\mathbf{u}_z = (v_{1,z}, \ldots, v_{d_1,z})$ and $\mathbf{u}_{i,j} = (v_{1,i,j}, \ldots, v_{d_1,i,j})$ for $z \in [w_2]$, $i \in [w_1]^+$, $j \in [n]$.

We now define matrices $\mathbf{B}_1, \ldots, \mathbf{B}_n$ and vectors $\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}, \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2}$ as follows. $\mathbf{B}_j$ has d_1 rows and $d_2 = w_1 + 1$ columns with the $(i+1)$th column being $\mathbf{u}_{i,j}^\top$ for $i = [w_1]^+$. Vector $\mathbf{s}_i$ is set to $\mathbf{e}_{i+1}$ for $i = [w_1]^+$, where $\mathbf{e}_i$ denotes the ith unit vector of size d_2, and $\hat{\mathbf{s}}_z$ is set to $\mathbf{u}_z$ for $z \in [w_2]$. These matrices and vectors depend only on $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{d_1}$, which in turn depends on Δ' only. The entries in Δ' are the coefficients of the monomials obtained by multiplying $r_{i'}$ with various ct-enc polynomials. Hence, they only depend on x and, in particular, not on y. Further, it is easy to observe that all the operations involved in computing $\mathbf{B}_j$, $\mathbf{s}_i$, $\hat{\mathbf{s}}_z$ are efficient. Thus, one can define two deterministic polynomial time algorithms EncB and EncS that on input x only, output $\mathbf{B}_1, \ldots, \mathbf{B}_n$ and $\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}$, $\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2}$ respectively.

We need to verify that if we substitute $\hat{s}_z$ with $\hat{\mathbf{s}}_z^\top$ and $s_i b_j$ with $\mathbf{B}_j \mathbf{s}_i^\top$ in any ct-enc polynomial c_ℓ, then we get an all zeroes vector. On performing such a substitution, we have

$$\sum_z \eta_{\ell,z} \mathbf{u}_z^\top \quad + \quad \sum_{i,j} \eta_{\ell,i,j} (\mathbf{u}_{0,j}^\top, \ldots, \mathbf{u}_{w_1,j}^\top) \mathbf{e}_{i+1}^\top \quad = \quad \sum_z \eta_{\ell,z} \mathbf{u}_z^\top \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \mathbf{u}_{i,j}^\top$$

The pth element in the column vector above is given by (2), which is equal to 0 for any p.

In the special case where Δ''s kernel is empty, $\mathbf{B}_1, \ldots, \mathbf{B}_n$ are all set to $d_1 \times d_2$ matrices with zero entries; $\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2}$ are set to the zero vector of size d_1; $\mathbf{s}_1, \ldots, \mathbf{s}_{w_1}$ are set to the zero vector of size d_2; and $\mathbf{s}_0$ is set to $(1, 0, \ldots, 0)$. It is easy to see that all ct-enc polynomials still evaluate to zero upon substitution.

does not satisfy a policy, i.e. the associated set of rows cannot be linearly combined to get a certain vector $\mathbf{v}$, then one can find a vector orthogonal to all those rows but not to $\mathbf{v}$. See, for instance, [Bei11, Claim 2] for a formal proof.

We also need to make sure that with the appropriate choice of vectors for the key-enc variables, all the key-enc polynomials also evaluate to the zero vector. Recall that such polynomials are given by

$$k_t \quad = \quad \phi_t \alpha \quad + \sum_{z' \in [m_2]} \phi_{t,z'} \hat{r}_{z'} \quad + \sum_{\substack{i' \in [m_1],\\ j \in [n]}} \phi_{t,i',j} r_{i'} b_j$$

for $t \in [m_3]$. When they are multiplied with a non-lone ct-enc variable s_i, we get the monomials αs_i, $s_i \hat{r}_{z'}$, $s_i r_{i'} b_j$ for $i \in [w_1]^+$ and i', j, z' as above.

Recall that the columns of Δ can be linearly combined using $\{\lambda_q\}_{q \in Q}$ to get the zero vector. Going back to the product of $r_{i'}$ with c_ℓ, we can say that

$$\sum_z \eta_{\ell,z} \lambda_{\hat{s}_z r_{i'}} \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \lambda_{s_i b_j r_{i'}} \quad = \quad 0$$

irrespective of what ℓ and i' are because only the entries in the columns associated with monomials $\hat{s}_z r_{i'}$, $s_i b_j r_{i'}$ are non-zero. Hence, the vector $\mathbf{w}_{i'}$ given by $(\lambda_{\hat{s}_1 r_{i'}}, \ldots, \lambda_{\hat{s}_{w_2} r_{i'}}, \lambda_{s_0 b_1 r_{i'}}, \ldots, \lambda_{s_0 b_n r_{i'}}, \ldots,$ $\lambda_{s_{w_1} b_1 r_{i'}}, \ldots, \lambda_{s_{w_1} b_n r_{i'}})$ lies in the kernel of Δ'. (Recall that no matter what key-enc non-lone variable is chosen, one always gets the same Δ'.) In other words, there exists a vector $\mathbf{r}_{i'}$ of size d_1 such that $[\mathbf{v}_1^\top, \ldots, \mathbf{v}_{d_1}^\top]\mathbf{r}_{i'}^\top = \mathbf{w}_{i'}$. Now the transpose of $\mathbf{r}_{i'} \mathbf{B}_j$ is given by

$$\begin{bmatrix} \mathbf{u}_{0,j} \\ \vdots \\ \mathbf{u}_{w_1,j} \end{bmatrix} \mathbf{r}_{i'}^\top \quad = \quad \begin{bmatrix} v_{1,0,j} & \cdots & v_{d_1,0,j} \\ \vdots & \vdots & \vdots \\ v_{1,w_1,j} & \cdots & v_{d_1,w_1,j} \end{bmatrix} \mathbf{r}_{i'}^\top \quad = \quad \begin{bmatrix} \lambda_{s_0 b_j r_{i'}} \\ \vdots \\ \lambda_{s_{w_1} b_j r_{i'}} \end{bmatrix}$$

for every $j \in [n]$. In the special case where Δ''s kernel is empty, set $\mathbf{r}_{i'}$ to be the zero vector of size d_1. The relation $\mathbf{r}_{i'} \mathbf{B}_j = (\lambda_{s_0 b_j r_{i'}}, \ldots, \lambda_{s_{w_1} b_j r_{i'}})$ for all j still holds because $\mathbf{w}_{i'}$ must be zero.

Define the remaining vectors as follows: $\mathbf{a}$ is set to be $[\lambda_{\alpha s_0}, \ldots, \lambda_{\alpha s_{w_1}}]$ and $\hat{\mathbf{r}}_{z'}$ to be $[\lambda_{s_0 \hat{r}_{z'}}, \ldots,$ $\lambda_{s_{w_1} \hat{r}_{z'}}]$ for $z' \in [m_2]$. (Note that the first element of $\mathbf{a}$ is not zero.) When we substitute α with $\mathbf{a}$, $\hat{r}_{z'}$ with $\hat{\mathbf{r}}_{z'}$ and $r_{i'} b_j$ with $\mathbf{r}_{i'} \mathbf{B}_j$ in k_t for $t \in [m_3]$, we get

$$\phi_t [\lambda_{\alpha s_0}, \ldots, \lambda_{\alpha s_{w_1}}] \quad + \quad \sum_{z'} \phi_{t,z'} [\lambda_{s_0 \hat{r}_{z'}}, \ldots, \lambda_{s_{w_1} \hat{r}_{z'}}] \quad + \quad \sum_{i',j} \phi_{t,i',j} [\lambda_{s_0 b_j r_{i'}}, \ldots, \lambda_{s_{w_1} b_j r_{i'}}].$$

The ith element of this sum is given by

$$\phi_t \lambda_{\alpha s_i} \quad + \quad \sum_{z'} \phi_{t,z'} \lambda_{s_i \hat{r}_{z'}} \quad + \quad \sum_{i',j} \phi_{t,i',j} \lambda_{s_i r_{i'} b_j}$$

for $i \in [w_1]^+$. It is easy to see that the above quantity is zero when we consider the row in Δ attached with the product $s_i k_t$.

One can define a deterministic polynomial time algorithm EncR that on input x and y, computes how the columns of Δ can be combined to get the zero vector, and then uses this information to define $\mathbf{a}$, $\hat{\mathbf{r}}_{z'}$, $\mathbf{r}_{i'}$ as shown above.

The proof for the co-selective symbolic property is analogous to the proof above, so we skip the details. □

5 Predicate Encryption from Pair Encodings

In this section, we describe how any pair encoding scheme for a predicate can be transformed into an encryption scheme for the same predicate in dual system groups (DSG), introduced by

Chen and Wee [CW14a], and later used and improved by several works [CGW15, AC16, Att16]. This transformation is a two-step process: first we *augment* an encoding so that it satisfies a few extra properties (Section 5.1)[10]; next we apply the transformation from Agrawal and Chase [AC16] (Section 5.4).

5.1 Augmenting Pair Encodings

We need the matrices and vectors involved in the symbolic property to have some extra features, so that we can prove the security of the derived predicate encryption scheme from our q-ratio assumption. Towards this, we show how any pair encoding scheme that satisfies Sym-Prop can be transformed into another scheme that satisfies a more constrained version of this property, with only a few additional variables and polynomials.

We note that, although they are presented monolithically, many of the pair encodings introduced by Attrapadung [Att14a] can be viewed as the result of applying a very similar augmentation to simpler underlying encodings. Thus, our results also help explain the structure of those previous encodings.

Recall that the algorithms of symbolic security output $\mathbf{a}$ for α, $\mathbf{B}_1, \ldots, \mathbf{B}_n$ for common variables, $\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}$ for non-lone ct-enc variables, and $\mathbf{r}_1, \ldots, \mathbf{r}_{m_1}$ for key-enc non-lone variables. Let $\mathbf{b}_j$ denote the first column of $\mathbf{B}_j$ and $s_{i,1}$ the first element of $\mathbf{s}_i$.

Definition 5.1 (Enhanced symbolic property)**.** *A pair encoding scheme satisfies* (d_1, d_2)-Sym-$\mathsf{Prop}^\star$ *for a predicate* P_κ *if it satisfies selective and co-selective* (d_1, d_2)-Sym-Prop *for* P_κ *but under the following constraints for both*

1. $\mathbf{a}$ *is set to* $(1, 0, \ldots, 0)$.
2. *In every* ct-enc *polynomial, if* $s_i b_j$ *is replaced by*
   - $\mathbf{s}_i^\top \mathbf{b}_j$ *then we get a matrix with non-zero elements in the first row only;*
   - $s_{i,1} \mathbf{B}_j$ *then we get a matrix with non-zero elements in the first column only.*

   (The lone variables are replaced by the zero vector.)
3. *In every* key-enc *polynomial, if we replace* $r_{i'} b_j$ *with* $\mathbf{b}_j^\top \mathbf{r}_{i'}$*, then we get a diagonal matrix. (The lone variables, once again, are replaced by the zero vector.)*
4. *The set of vectors* $\{\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}\}$ *is linearly independent, and so is the set* $\{\mathbf{r}_1, \ldots, \mathbf{r}_{m_1}\}$.

We convert any pair encoding that satisfies Sym-Prop into one that satisfies Sym-Prop* in three steps. First we show that with only one additional key-enc non-lone variable, an additional common variable, and an extra ct-enc polynomial, we can get an encoding scheme for which the vector $\mathbf{a}$ corresponding to α can be set to $(1, 0, \ldots, 0)$ (in proving that Sym-Prop holds). Next, with two extra common variables, and an additional variable and a polynomial each in the ciphertext and key encoding, one can satisfy the second and third properties from above. Finally, a simple observation can be used to satisfy the fourth property as well.

Theorem 5.2 (Augmentation)**.** *Suppose a* PES *for a predicate family* $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ *outputs* n *on input* par, $(w_1, w_2, \mathbf{c})$ *on input* $x \in \mathcal{X}_\kappa$, $(m_1, m_2, \mathbf{k})$ *on input* $y \in \mathcal{Y}_\kappa$ *and satisfies* (d_1, d_2)-Sym-Prop, *then there exists another* PES *for* P_κ *that outputs* $n+3$ *on input* par, $(w_1+1, w_2, \overline{\mathbf{c}})$ *on input* x *and*

[10] This step need not be applied if the properties are already satisfied.

$(m_1+2, m_2, \overline{\mathbf{k}})$ *on input* y, *where* $|\overline{\mathbf{c}}| = |\mathbf{c}|+2$ *and* $|\overline{\mathbf{k}}| = |\mathbf{k}|+1$, *and satisfies* $(\max(d_1, d_2-1)+M_1+1, d_2+W_1+2)$-Sym-Prop*, *where* M_1 *and* W_1 *are bounds on the number of* key-enc *and* ct-enc *non-lone variables, respectively.*[11]

Proof. Let $\Gamma' = (\mathsf{Param}', \mathsf{EncCt}', \mathsf{EncKey}', \mathsf{Pair}')$ be a PES that satisfies (d_1, d_2)-Sym-Prop w.r.t. the triple of algorithms $(\mathsf{EncB}', \mathsf{EncS}', \mathsf{EncR}')$. Without loss of generality, we can assume that $d_1 \geq d_2 - 1$ (see the discussion that follows Definition 3.1).

First transformation. We build a new encoding scheme Γ as follows. Introduce a new common variable b and a non-lone key-enc variable r. In every key-enc polynomial, α' is replaced with $\alpha' + rb$. A new polynomial, $s'_0 b$, is added to the ciphertext encoding. It is easy to see that the correctness still holds: using the matrices output by Pair' we get $(\alpha' + rb)s'_0 = \alpha' s'_0 + rbs'_0$, from which the second term can be removed by multiplying $s'_0 b$ by r.

To prove the co-selective property for the new scheme, we exploit the same property of Γ'. Let $\mathbf{V}$ be a d_2 dimensional square matrix such that $\mathbf{a}'\mathbf{V} = (1, 0, \ldots, 0)$, where $\mathbf{a}'$ is output by $\mathsf{EncR}(y)$ as the vector for α'. The new algorithms we define, i.e. EncB, EncS and EncR, can all compute $\mathbf{V}$ since they have access to y. EncR outputs the vectors that EncR$'$ does but with all the lone variables multiplied by $\mathbf{V}$ on the right. EncB outputs the matrices produced by EncB$'$ but after multiplying them with $\mathbf{V}$ on the right. It also outputs a zero matrix for the new common variable b. Finally, EncS outputs the vectors that EncS$'$ does but with all the non-lone variables multiplied by the transpose of $\mathbf{V}^{-1}$ on the right (so that when we take the transpose of the new vectors, $\mathbf{V}^{-1}$ is multiplied on the left). It is easy to check now that all the polynomials still evaluate to zero. Also, the new vector for α' and s'_0 are $\mathbf{a}'\mathbf{V}$ and $\mathbf{s}'_0(\mathbf{V}^{-1})^\top$, respectively. Since $\langle \mathbf{a}', \mathbf{s}'_0 \rangle \neq 0$, so is the inner product of the new vectors.

Turning to the selective property, let $\mathbf{U}$ be a d_2 dimensional square matrix such that $\mathbf{s}'_0\mathbf{U}^\top = (1, 0, \ldots, 0)$, where $\mathbf{s}'_0$ is output by $\mathsf{EncS}(x)$ as the vector for s'_0. In the case of selective property, all the algorithms have access to x, hence they can compute $\mathbf{U}$. Now, EncS outputs the vectors that EncS$'$ does but with all the non-lone variables multiplied by $\mathbf{U}^\top$ on the right. EncB outputs the matrices produced by EncB$'$ but after multiplying them with $\mathbf{U}^{-1}$ on the right. It also outputs a matrix $\mathbf{B}$ for the new variable b with $B_{i,i+1} = 1$ for $i \in [d_2 - 1]$ and zero every where else (this is why we need $d_1 \geq d_2 - 1$). In particular, the first column of $\mathbf{B}$ does not have any non-zero entry. The ciphertext polynomials borrowed from Γ' still evaluate to zero, as can be easily seen. The new polynomial $s'_0 b$ upon substitution gives $\mathbf{B}\mathbf{U}\mathbf{s}'^\top_0$, which is equal to zero.

Let $\mathbf{a}'\mathbf{U}^{-1} = (\lambda_1, \ldots, \lambda_{d_2})$, where λ_1 cannot be zero because $\mathbf{s}'_0\mathbf{U}^\top = (1, 0, \ldots, 0)$ and $\langle \mathbf{a}', \mathbf{s}'_0 \rangle \neq 0$. EncR runs EncR$'$ to obtain vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_{m_1}$, $\mathbf{a}'$, $\hat{\mathbf{r}}'_1, \ldots, \hat{\mathbf{r}}'_{m_2}$. It outputs $1/\lambda_1\mathbf{r}'_1, \ldots, 1/\lambda_1\mathbf{r}'_{m_1}$, $(1, 0, \ldots, 0)$, $1/\lambda_1\hat{\mathbf{r}}'_1\mathbf{U}^{-1}, \ldots, 1/\lambda_1\hat{\mathbf{r}}'_{m_2}\mathbf{U}^{-1}$, along with a vector $\mathbf{r} = (\lambda_2/\lambda_1, \ldots, \lambda_{d_2}/\lambda_1, 0, \ldots, 0)$ for the new variable r. Observe that $\alpha' + rb$ upon substitution gives $(1, \lambda_2/\lambda_1, \ldots, \lambda_{d_2}/\lambda_1)$, which is equal to $1/\lambda_1\mathbf{a}'\mathbf{U}^{-1}$. Hence, when we replace the variables in a key-enc polynomial with the new vectors/matrices, we can factor out $1/\lambda_1\mathbf{U}^{-1}$ on the right. Thus, all the polynomials continue to evaluate to zero.

Second transformation. This transformation can be directly applied to any pair encoding scheme that satisfies symbolic property with $\mathbf{a}$ set to $(1, 0, \ldots, 0)$. Let $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ be one such scheme with $(\mathsf{EncB}, \mathsf{EncS}, \mathsf{EncR})$ as the algorithms for symbolic security. We augment again to build Γ^* as follows:

[11] As we will see later, when a pair encoding scheme is transformed into a predicate encryption scheme, the parameters of Sym-Prop* have no effect on the construction. They only affect the size of assumption on which the security of encryption scheme is based.

- $\mathsf{Param}^*(\mathsf{par}) \to n+2$, where $n = \mathsf{Param}(\mathsf{par})$. Let $\mathbf{b}$ denote the vector $(b_1, \dots, b_n)$ and $\mathbf{b}^*$ denote $\mathbf{b} \circ (b^*, b^{**})$.

- $\mathsf{EncCt}^*(x, N)$. Run $\mathsf{EncCt}(x, N)$ to obtain a vector $\mathbf{c}$ in variables $\mathbf{s} = (s_0, \dots, s_{w_1})$, $\hat{\mathbf{s}}$ and $\mathbf{b}$. Output $\mathbf{c}^* = (\mathbf{c}, s^* b^* + s_0 b^{**})$ as the new vector of polynomials, where s^* is a new variable. Hence the polynomials in $\mathbf{c}^*$ are in variables $\mathbf{s}^* = (s^*, \mathbf{s})$, $\hat{\mathbf{s}}$ and $\mathbf{b}^*$.

- $\mathsf{EncKey}^*(y, N)$. Run $\mathsf{EncKey}(y, N)$ to obtain a vector $\mathbf{k}$ in variables $\mathbf{r}$, $\hat{\mathbf{r}} = (\alpha, \dots)$ and $\mathbf{b}$. Output $\mathbf{k}^* = (\mathbf{k}, \alpha + r^* b^*)$ as the new vector of polynomials, where r^* is a new variable, with α replaced by $r^* b^{**}$ in all the polynomials in $\mathbf{k}$. One can see that the polynomials in $\mathbf{k}^*$ are in variables $\mathbf{r}^* = (r^*, \mathbf{r})$, $\hat{\mathbf{r}}$ and $\mathbf{b}^*$.

- $\mathsf{Pair}^*(x, y, N)$. Run Pair to get matrices $\mathbf{E}$ and $\overline{\mathbf{E}}$. Define $\mathbf{E}^*$ by setting $E^*_{0,m_3+1} = 1$, $E^*_{i,t} = E_{i-1,t}$ for $i \in [w_1 + 1]$, $t \in [m_3]$, and 0 at every other position. Also, define $\overline{\mathbf{E}}^*$ by setting $\overline{E}^*_{w_3+1,1} = -1$, $\overline{E}^*_{\ell,i'+1} = \overline{E}_{\ell,i'}$ for $\ell \in [w_3]$, $i' \in [m_1]$, and 0 everywhere else.

Correctness holds because

$$\begin{aligned}
\mathbf{s}^* \mathbf{E}^* (\mathbf{k}^*)^\top + \mathbf{c}^* \overline{\mathbf{E}}^* (\mathbf{r}^*)^\top &= s^*(\alpha + r^* b^*) + \sum_{i,t} s_i E_{i,t} k_t - (s^* b^* + s_0 b^{**}) r^* + \sum_{\ell,i'} c_\ell \overline{E}_{\ell,i'} r_{i'} \\
&= s^*(\alpha + r^* b^*) - (s^* b^* + s_0 b^{**}) r^* + s_0 r^* b^{**} \quad = \quad \alpha s^*
\end{aligned}$$

We define new algorithms $(\mathsf{EncB}^*, \mathsf{EncS}^*, \mathsf{EncR}^*)$ to show that $\mathsf{Sym}\text{-}\mathsf{Prop}$ continues to hold (both selective and co-selective properties will be proved at the same time). Let $\mathbf{E}_{i,j}$ be a $d_1 \times (d_2 + 1)$ matrix with 1 in the ith row and jth column, and 0 everywhere else. Also, let $\overline{\mathbf{e}}_i$ be a unit vector of size $d_2 + 1$ with 1 at the ith position, and $\mathbf{e}_j$ be a unit vector of size d_1 with 1 at the jth position.

- EncB^* outputs an extended matrix $\overline{\mathbf{B}}$ for every matrix $\mathbf{B}$ output by EncB, where $\mathbf{B}^*$ is just $\mathbf{B}$ with a column of zeroes added to the left. It also outputs $\mathbf{B}^* = -\mathbf{E}_{1,1}$ and $\mathbf{B}^{**} = \mathbf{E}_{1,2}$ corresponding to b^* and b^{**}, respectively.

- EncS^* runs EncS to obtain vectors $\mathbf{s}_0, \dots, \mathbf{s}_{w_1}$, $\hat{\mathbf{s}}_1, \dots, \hat{\mathbf{s}}_{w_2}$. Suppose $\mathbf{s}_0$ is given by $(\lambda_1, \dots, \lambda_{d_2})$ where $\lambda_1 \neq 0$. The vector $\mathbf{s}_i$ for a non-lone variable s_i is extended by adding a zero to the left, giving a new vector $\overline{\mathbf{s}}_i$. Output is $\overline{\mathbf{s}}_0, \dots, \overline{\mathbf{s}}_{w_1}$, $\hat{\mathbf{s}}_1, \dots, \hat{\mathbf{s}}_{w_2}$, along with a vector $\mathbf{s}^* = \lambda_1 \overline{\mathbf{e}}_1$ for the new variable s^*.

- EncR^* runs EncR to obtain vectors $\mathbf{r}_1, \dots, \mathbf{r}_{m_1}$, $\mathbf{a}$, $\hat{\mathbf{r}}_1, \dots, \hat{\mathbf{r}}_{m_2}$, where $\mathbf{a} = (1, 0, \dots, 0)$. The vector $\hat{\mathbf{r}}_z$ for a lone variable $\hat{r}_z$ is extended by appending a zero to the left, giving a new vector $\overline{\mathbf{r}}_z$, for $z \in [m_2]$. Output is $\mathbf{r}_1, \dots, \mathbf{r}_{m_1}, \overline{\mathbf{e}}_1, \overline{\mathbf{r}}_1, \dots, \overline{\mathbf{r}}_{m_2}$, along with a vector $\mathbf{r}^* = \mathbf{e}_1$ for the new variable r^*.

Recall that we replaced α by $r^* b^{**}$ in the polynomials in $\mathbf{k}$. Now, when we substitute $r^* b^{**}$ by $\mathbf{e}_1 \mathbf{E}_{1,2}$, we get $\overline{\mathbf{e}}_2$, or $\mathbf{a} = (1, 0, \dots, 0)$ with a zero inserted at the leftmost position. Hence, the polynomials in $\mathbf{k}$ still evaluate to zero. It is also easy to see that when the polynomials in $\mathbf{c}$ are evaluated with the extended matrices and vectors, they evaluate to zero as before. We now argue about the two new polynomials we added. The vector $\overline{\mathbf{s}}_0$ can be written as $\sum_{i=1}^{d_2} \lambda_i \overline{\mathbf{e}}_{i+1}$. Hence, $s^* b^* + s b^{**}$ evaluates to $-\lambda_1 \mathbf{E}_{1,1} \overline{\mathbf{e}}_1^\top + \mathbf{E}_{1,2}(\lambda_1 \overline{\mathbf{e}}_2^\top + \dots) = -\lambda_1 \mathbf{e}_1^\top + \lambda_1 \mathbf{e}_1^\top = \mathbf{0}^\top$. Further, $\alpha + r^* b^*$ evaluates to $\overline{\mathbf{e}}_1 - \mathbf{e}_1 \mathbf{E}_{1,1} = \mathbf{0}$.

We now show how the second and third properties in Definition 5.1 are realized by the new matrices and vectors. To see why the first point in the second property and the third property are satisfied, observe that there is only one matrix, $\mathbf{B}^* = -\mathbf{E}_{1,1}$, with a non-zero value in the first column. Moreover, the corresponding common variable b^* is used in only one ct-enc polynomial $s^*b^* + s_0 b^{**}$. After substitution, we get $\lambda_1 \bar{\mathbf{e}}_1^\top \cdot (-\mathbf{e}_1) = -\lambda_1 \mathbf{E}_{1,1}^\top$. Further, only $\alpha + r^* b^*$ involves b^* in the case of key-enc polynomials. And after substitution we have $-\mathbf{e}_1^\top \cdot \mathbf{e}_1$, which is a diagonal matrix with -1 in the first row and column, and 0 elsewhere. For the second point in the second property, note that only $\mathbf{s}^*$'s first element has a non-zero value among the non-lone variables, and it appears in only one polynomial $s^* b^* + s_0 b^{**}$. Upon substitution, we just have $-\lambda_1 \mathbf{E}_{1,1}$.

Third transformation. Suppose an encoding scheme satisfies Sym-Prop while meeting the first three conditions of Definition 5.1. Consider the vectors $\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}$ for non-lone ct-enc variables. If we modify the output of the three algorithms involved in the symbolic security proof slightly as follows:

- add a vector of size $w_1 + 1$ with 1 at the ith position to the right of $\mathbf{s}_i$ for $i \in [w_1]^+$,
- add a zero matrix of size $d_1 \times (w_1 + 1)$ to the right of $\mathbf{B}_j$ for $j \in [n]$, and
- add a zero vector of size $w_1 + 1$ to the right of $\mathbf{a}$ and $\hat{\mathbf{r}}_z$ for $z \in [m_2]$,

then it is easy to verify that Sym-Prop (as well as the first three properties of Definition 5.1) still holds, and additionally, $\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}$ now form an independent set of vectors. In an analogous manner, we can also make the vectors $\mathbf{r}_1, \ldots, \mathbf{r}_{m_1}$ independent. (Note that the modifications needed in this case are complementary to the ones described above for the case of ct-enc variables.) □

We now prove three combinatorial facts that follow from the additional properties of augmented pair encoding schemes. To keep the presentation simple, we use minimal notation to prove our point. All the lemmas hold for any range of i, j, any integers $\mu_{i,j}$, and any vectors $\mathbf{s}_i$ and matrices $\mathbf{B}_j$ over integers.

Lemma 5.3. *If $\sum_{i,j} \mu_{i,j} \mathbf{s}_i^\top \mathbf{b}_j$ is a matrix with non-zero values in the first row only, then for all matrices $\mathbf{U}$, the value of $\mathbf{m}^\top := \sum_{i,j} \mu_{i,j} \mathbf{B}_j \mathbf{U} \mathbf{s}_i^\top$ does not depend on the first row of $\mathbf{U}$ if the element in $\mathbf{U}$'s first row and column is 0, where $\mathbf{b}_j$ is the first column of $\mathbf{B}_j$.*

Proof. If $s_{i,q}$ denotes the qth element of $\mathbf{s}_i$, then we know that if $q \neq 1$, $\sum_{i,j} \mu_{i,j} \mathbf{b}_j s_{i,q} = 0$. The product $\mathbf{B}_j \mathbf{U}$ in $\mathbf{m}$ can be written as a sum of matrices, where the ℓth matrix is a product of the ℓth column of $\mathbf{B}_j$ and the ℓth row of $\mathbf{U}$. Since we are concerned with the first row of $\mathbf{U}$ only, it is sufficient to consider the sum $\mathbf{n}^\top := \sum_{i,j} \mu_{i,j} \mathbf{b}_j^\top \mathbf{u} \mathbf{s}_i^\top$, where $\mathbf{u}$ denotes the first row of $\mathbf{U}$. Now, $\mathbf{n} = \sum_{i,j} \mu_{i,j} \mathbf{b}_j \sum_q u_q s_{i,q}$, where u_q is the qth element of $\mathbf{u}$. But since $u_1 = 0$, we can rewrite $\mathbf{n}$ as $\sum_{q \neq 1} u_q \sum_{i,j} \mu_{i,j} \mathbf{b}_j s_{i,q}$. The latter sum is $\mathbf{0}$ whenever $q \neq 1$, hence $\mathbf{n} = \mathbf{0}$. □

Lemma 5.4. *If $\sum_{i,j} \mu_{i,j} s_{i,1} \mathbf{B}_j$ is a matrix with non-zero values in the first column only, then for all matrices $\mathbf{U}$, the value of $\mathbf{m}^\top := \sum_{i,j} \mu_{i,j} \mathbf{B}_j \mathbf{U} \mathbf{s}_i^\top$ does not depend on the first column of $\mathbf{U}$ if the element in $\mathbf{U}$'s first row and column is 0, where $s_{i,1}$ is the first element of $\mathbf{s}_i$.*

Proof. The product $\mathbf{U}\mathbf{s}_i^\top$ can be written as a linear combination of the columns of $\mathbf{U}$ where the coefficients are the elements of $\mathbf{s}_i$. Since we are concerned with the first column only, it is sufficient to consider $\mathbf{n}^\top := \sum_{i,j} \mu_{i,j} \mathbf{B}_j \mathbf{u}^\top s_{i,1}$, where $\mathbf{u}$ is the first column of $\mathbf{U}$ and $s_{i,1}$ is the first element of $\mathbf{s}_i$. Rewriting $\mathbf{n}^\top$ as $\left(\sum_{i,j} \mu_{i,j} s_{i,1} \mathbf{B}_j\right) \mathbf{u}^\top$, we can easily see that it has to be $\mathbf{0}$: again, we can express this as a linear combination of the columns of $\sum_{i,j} \mu_{i,j} s_{i,1} \mathbf{B}_j$ using as coefficients the elements of $\mathbf{u}$; all columns except the first are zero by assumption, and the first column does not matter because the first element of $\mathbf{u}$ is 0. □

Lemma 5.5. *If $\sum_{i,j}\mu_{i,j}\mathbf{b}_j^\top\mathbf{r}_i$ is a matrix with non-zero elements in the diagonal only, then for all matrices $\mathbf{V}$ which have zeroes in the diagonal, the first element of the vector $\mathbf{m} := \sum_{i,j}\mu_{i,j}\mathbf{r}_i\mathbf{V}\mathbf{B}_j$ is zero, where $\mathbf{b}_j$ is the first column of $\mathbf{B}_j$.*

Proof. If $r_{i,p}$ and $b_{j,q}$ denote the pth and qth elements of $\mathbf{r}_i$ and $\mathbf{b}_j$ respectively, then we know that if $p \neq q$, $\sum_{i,j}\mu_{i,j}r_{i,p}b_{j,q} = 0$. Since the first element of $\mathbf{m}$ depends on the first column of $\mathbf{B}_j$ only, it is sufficient to consider the sum $n := \sum_{i,j}\mu_{i,j}\mathbf{r}_i\mathbf{V}\mathbf{b}_j^\top = \sum_{i,j}\mu_{i,j}\sum_{p,q}r_{i,p}v_{p,q}b_{j,q}$, where $v_{p,q}$ is the (p,q)th element of $\mathbf{V}$. But since $\mathbf{V}$ has zeroes on the diagonal, we can rewrite n as $\sum_{p\neq q}v_{p,q}\sum_{i,j}\mu_{i,j}r_{i,p}b_{j,q}$. The latter sum is 0 whenever $p \neq q$, hence $n = 0$. □

5.2 Dual System Groups

Dual system groups (DSG) were introduced by Chen and Wee [CW14a] and generalized by Agrawal and Chase [AC16]. The latter work also shows that the two instantiations of DSG – in composite-order groups under the subgroup decision assumption and in prime-order groups under the decisional linear assumption – given by Chen and Wee satisfy the generalized definition as well.

The following definition of dual system groups has been taken almost verbatim from Agrawal and Chase [AC16]. It is parameterized by a security parameter λ and a number n, and consists of six PPT algorithms:

- $\mathsf{SampP}(1^\lambda, 1^n)$: On input 1^λ and 1^n, SampP outputs public parameters PP and secret parameters SP, which have the following properties:
  - PP contains a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, a homomorphism μ from $\mathbb{H}$ to $\mathbb{G}_T$, along with some additional parameters used by SampG, SampH. Given PP, we know the exponent of group $\mathbb{H}$ and how to sample uniformly from it; let $N = \exp(\mathbb{H})$. It is required that N is a product of distinct primes of $\Theta(\lambda)$ bits.
  - SP contains $\tilde{h} \in \mathbb{H}$ (where $\tilde{h} \neq 1_{\mathbb{H}}$) along with additional parameters used by $\overline{\mathsf{SampG}}$ and $\overline{\mathsf{SampH}}$.
- SampGT takes an element in the image of μ and outputs another element from $\mathbb{G}_T$.
- SampG and SampH take PP as input and output a vector of $n+1$ elements from $\mathbb{G}$ and $\mathbb{H}$ respectively.
- $\overline{\mathsf{SampG}}$ and $\overline{\mathsf{SampH}}$ take both PP and SP as inputs and output a vector of $n+1$ elements from $\mathbb{G}$ and $\mathbb{H}$ respectively.

Properties

All the properties below hold for every PP and SP output by SampP. Let SampG_0 be the algorithm that outputs only the first element of SampG. Analogously, SampH_0, $\overline{\mathsf{SampG}_0}$ and $\overline{\mathsf{SampH}_0}$ can be defined. A dual system group is *correct* if it satisfies the following two properties:

Projective: For all $h \in \mathbb{H}$ and coin tosses σ, $\mathsf{SampGT}(\mu(h);\sigma) = e(\mathsf{SampG}_0\,(\mathrm{PP};\sigma), h)$.

Associative: If $(g_0, g_1, \ldots, g_n)$ and $(h_0, h_1, \ldots, h_n)$ are samples from $\mathsf{SampG}(\mathrm{PP})$ and $\mathsf{SampH}(\mathrm{PP})$ respectively, then for all $i \in [1, n]$, $e(g_0, h_i) = e(g_i, h_0)$.

For *security*, the following three properties should hold:

Orthogonality: $\mu(\tilde{h}) = 1_{\mathbb{G}_T}$.

Non-degeneracy:

1. $\overline{\mathsf{SampH}}_0(\textsc{pp},\textsc{sp}) \cong \tilde{h}^{\delta}$, where $\delta \leftarrow_R \mathbb{Z}_N$.
2. $\exists\, \tilde{g} \in \mathbb{G}$ s.t. $\tilde{g} \neq 1_{\mathbb{G}}$ and $\overline{\mathsf{SampG}}_0(\textsc{pp},\textsc{sp}) \cong \tilde{g}^{\alpha}$, where $\alpha \leftarrow_R \mathbb{Z}_N$.
3. For all $\hat{g}_0 \leftarrow \overline{\mathsf{SampG}}_0(\textsc{pp},\textsc{sp})$, $e(\hat{g}_0,\tilde{h})^{\beta}$ is uniformly distributed over $\mathbb{G}_T$, where $\beta \leftarrow_R \mathbb{Z}_N$.

Indistinguishability. For two (positive) polynomials p and q, define $\mathbf{G}$, $\mathbf{H}$, $\hat{\mathbf{G}}$, $\hat{\mathbf{H}}$, $\hat{\mathbf{G}}'$, $\hat{\mathbf{H}}'$ as follows:

$$(\textsc{pp},\textsc{sp}) \leftarrow \mathsf{SampP}(1^{\lambda},1^{n}); \quad \gamma_1,\gamma_2,\ldots,\gamma_n \leftarrow_R \mathbb{Z}_N;$$

$$\mathbf{g}_1,\mathbf{g}_2,\ldots,\mathbf{g}_{p(\lambda)} \leftarrow \mathsf{SampG}(\textsc{pp}); \mathbf{G} := (\mathbf{g}_1,\mathbf{g}_2,\ldots,\mathbf{g}_{p(\lambda)});$$
$$\mathbf{h}_1,\mathbf{h}_2,\ldots,\mathbf{h}_{q(\lambda)} \leftarrow \mathsf{SampH}(\textsc{pp}); \mathbf{H} := (\mathbf{h}_1,\mathbf{h}_2,\ldots,\mathbf{h}_{q(\lambda)});$$
$$\forall\, i \in [p(\lambda)], \quad \hat{\mathbf{g}}_i := (\hat{g}_{i,0},\ldots) \leftarrow \overline{\mathsf{SampG}}(\textsc{pp},\textsc{sp}); \quad \hat{\mathbf{g}}_i' := (1,\hat{g}_{i,0}^{\gamma_1},\hat{g}_{i,0}^{\gamma_2},\ldots,\hat{g}_{i,0}^{\gamma_n})$$
$$\forall\, j \in [q(\lambda)], \quad \hat{\mathbf{h}}_j := (\hat{h}_{j,0},\ldots) \leftarrow \overline{\mathsf{SampH}}(\textsc{pp},\textsc{sp}); \quad \hat{\mathbf{h}}_j' := (1,\hat{h}_{j,0}^{\gamma_1},\hat{h}_{j,0}^{\gamma_2},\ldots,\hat{h}_{j,0}^{\gamma_n})$$
$$\hat{\mathbf{G}} := (\hat{\mathbf{g}}_1,\hat{\mathbf{g}}_2,\ldots,\hat{\mathbf{g}}_{p(\lambda)}); \hat{\mathbf{H}} := (\hat{\mathbf{h}}_1,\hat{\mathbf{h}}_2,\ldots,\hat{\mathbf{h}}_{q(\lambda)});$$
$$\hat{\mathbf{G}}' := (\hat{\mathbf{g}}_1',\hat{\mathbf{g}}_2',\ldots,\hat{\mathbf{g}}_{p(\lambda)}'); \hat{\mathbf{H}}' := (\hat{\mathbf{h}}_1',\hat{\mathbf{h}}_2',\ldots,\hat{\mathbf{h}}_{q(\lambda)}').$$

A dual system group is *Left Subgroup Indistinguishable*, *Right Subgroup Indistinguishable* and *Parameter hiding* if for all polynomials $p(\cdot)$ and $q(\cdot)$,

$$\{\textsc{pp},\mathbf{G}\} \approx \{\textsc{pp},\mathbf{G}\cdot\hat{\mathbf{G}}\}, \tag{3}$$
$$\{\textsc{pp},\tilde{h},\mathbf{G}\cdot\hat{\mathbf{G}},\mathbf{H}\} \approx \{\textsc{pp},\tilde{h},\mathbf{G}\cdot\hat{\mathbf{G}},\mathbf{H}\cdot\hat{\mathbf{H}}\}, \text{and} \tag{4}$$
$$\{\textsc{pp},\tilde{h},\hat{\mathbf{G}},\hat{\mathbf{H}}\} \equiv \{\textsc{pp},\tilde{h},\hat{\mathbf{G}}\cdot\hat{\mathbf{G}}',\hat{\mathbf{H}}\cdot\hat{\mathbf{H}}'\} \tag{5}$$

hold, respectively. Observe that the two distributions in (3) and (4) are computationally indistinguishable, while the two distributions in (5) are identical.

Additional property. Additionally, we require that there exists a way to sample the set-up parameters so that one not only gets PP and SP, but also some trapdoor information td that can be used to generate samples from $\overline{\mathsf{SampG}}$ and $\overline{\mathsf{SampH}}$ given only the first element. We formalize this property with the help of three algorithms SampP^*, $\mathsf{ExtendG}$ and $\mathsf{ExtendH}$.

- $\mathsf{SampP}^*(1^{\lambda},1^{n})$. On input 1^{λ} and 1^{n}, SampP outputs public parameters PP, secret parameters SP (with the same properties as that output by SampP), and a trapdoor td.
- $\mathsf{ExtendG}$ takes a $g \in \mathbb{G}$, PP, SP and td as inputs, and outputs an element in $\mathbb{G}^n$.
- $\mathsf{ExtendH}$ takes a $h \in \mathbb{H}$, PP, SP and td as inputs, and outputs an element in $\mathbb{H}^n$.

For every $(\textsc{pp},\textsc{sp},\mathsf{td})$ we want:

- For all $\hat{g} \leftarrow \overline{\mathsf{SampG}}_0(\textsc{pp},\textsc{sp})$, the distribution $(\hat{g},\mathsf{ExtendG}(\textsc{pp},\textsc{sp},\mathsf{td},\hat{g}))$ is identical to that produced by $\overline{\mathsf{SampG}}(\textsc{pp},\textsc{sp})$, conditioned on the first element being $\hat{g}$.
- For all $\hat{h} \leftarrow \overline{\mathsf{SampH}}_0(\textsc{pp},\textsc{sp})$, the distribution $(\hat{h},\mathsf{ExtendH}(\textsc{pp},\textsc{sp},\mathsf{td},\hat{h}))$ is identical to that produced by $\overline{\mathsf{SampH}}(\textsc{pp},\textsc{sp})$, conditioned on the first element being $\hat{h}$.

These algorithms are needed to prove the security of our construction, not its correctness, so they are best viewed as extra security conditions for a DSG.

It is easy to see that the two instantiations of Chen and Wee [CW14a] also have the additional property we desire. In their composite-order construction (Section 5.2), td is the vector $\mathbf{w}$, which is chosen at random from $\mathbb{Z}_N^n$. $\mathsf{ExtendG}$ on input g outputs $g^{\mathbf{w}}$, and $\mathsf{ExtendH}$ on input h outputs $h^{\mathbf{w}} \cdot \mathbf{X}_3$, where $\mathbf{X}_3 \leftarrow_R G_{p_3}^n$ (using PP). The prime-order construction (Section 6.2) is based on an asymmetric bilinear map $(N, \mathcal{G}, \mathcal{H}, \mathcal{G}_T, g, h, e)$. The groups $\mathbb{G}$ and $\mathbb{H}$ are defined to be $\mathcal{G}^{d+1}$ and $\mathcal{H}^{d+1}$, respectively, where d is the parameter in the linear assumption. The trapdoor td is given by $(\mathbf{f}, \mathbf{f}_1, \ldots, \mathbf{f}_n, \mathbf{f}^*, \mathbf{f}_1^*, \ldots, \mathbf{f}_n^*)$. When $\mathsf{ExtendG}$ gets $\mathbf{g} = (g_1, \ldots, g_{d+1}) \in \mathcal{G}^{d+1}$ as input, it first checks if $\mathbf{f} = \mathbf{0}$ or not. If it is, then it picks $\hat{s} \leftarrow_R \mathbb{Z}_p^*$; otherwise, $g^{\hat{s}}$ is computed by picking any f_i that is non-zero and raising g_i to f_i^{-1}. Then the output is set to $(g^{\hat{s}\mathbf{f}_1}, \ldots, g^{\hat{s}\mathbf{f}_n})$. $\mathsf{ExtendH}$ behaves in a similar way except that it uses $\mathbf{f}^*, \mathbf{f}_1^*, \ldots, \mathbf{f}_n^*$.

5.3 New Computational Assumption

We introduce a new assumption, called $\mathsf{q\text{-}ratio_{dsg}}$, on dual system groups parameterized by positive integers d_1 and d_2.

Definition 5.6 ((d_1, d_2)-$\mathsf{q\text{-}ratio_{dsg}}$ assumption)**.** *Consider the following distribution on a dual system group's elements:*

$$\mathsf{dsg\text{-}par} := (\mathrm{PP}, \mathrm{SP}, \mathsf{td}) \leftarrow \mathsf{SampP}^*(1^\lambda, 1^n);$$

$$\hat{g} \leftarrow \overline{\mathsf{SampG}}_0(\mathrm{PP}, \mathrm{SP}); \quad \hat{h} \leftarrow \overline{\mathsf{SampH}}_0(\mathrm{PP}, \mathrm{SP})$$

$$u_0, u_1, \ldots, u_{d_2}, v_1, \ldots, v_{d_1} \leftarrow_R \mathbb{Z}_N^*;$$

$$D_{\mathbb{G}} \quad := \quad \{\hat{g}^{u_i}\}_{i \in [d_2]^+} \quad \cup \quad \left\{ \hat{g}^{\frac{u_i}{u_j v_k}} \right\}_{i,j \in [d_2], i \neq j, k \in [d_1]};$$

$$D_{\mathbb{H}} \quad := \quad \{\hat{h}^{v_i}\}_{i \in [d_1]} \quad \cup \quad \left\{ \hat{h}^{\frac{v_i}{v_j u_k}} \right\}_{i,j \in [d_1], i \neq j, k \in [d_2]};$$

$$T_0 := \hat{h}^{1/u_0}; \quad T_1 \leftarrow \overline{\mathsf{SampH}}_0(\mathrm{PP}, \mathrm{SP}).$$[12]

We say that the (d_1, d_2)-$\mathsf{q\text{-}ratio_{dsg}}$ *assumption holds if for any* PPT *algorithm* $\mathcal{A}$*,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{qr_{dsg}}}(\lambda) := \left|\Pr[\mathcal{A}(1^\lambda, \mathsf{dsg\text{-}par}, D_{\mathbb{G}}, D_{\mathbb{H}}, T_0) = 1]\right| - \left|\Pr[\mathcal{A}(1^\lambda, \mathsf{dsg\text{-}par}, D_{\mathbb{G}}, D_{\mathbb{H}}, T_1) = 1]\right|$$

is negligible in λ*.*

Note that u_0 is present in exactly one of the terms in $D_{\mathbb{G}}$ and not at all in $D_{\mathbb{H}}$.

We also define a similar assumption on bilinear maps.

Definition 5.7 ((d_1, d_2)-$\mathsf{q\text{-}ratio}$ assumption)**.** *Consider the following distribution:*

$$\mathsf{par} := (N, \mathcal{G}, \mathcal{H}, \mathcal{G}_T, g, h, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$$

$$\hat{g} \leftarrow_R \mathcal{G}; \quad \hat{h} \leftarrow_R \mathcal{H}; \qquad u_0, u_1, \ldots, u_{d_2}, v_1, \ldots, v_{d_1} \leftarrow_R \mathbb{Z}_N^*;$$

$$D_{\mathcal{G}} \quad := \quad \{\hat{g}^{u_i}\}_{i \in [d_2]^+} \quad \cup \quad \left\{ \hat{g}^{\frac{u_i}{u_j v_k}} \right\}_{i,j \in [d_2], i \neq j, k \in [d_1]};$$

[12] There is a typo in the camera-ready version here. T_1 is a fresh sample from $\overline{\mathsf{SampH}}_0(\mathrm{PP}, \mathrm{SP})$, not a random element in $\mathbb{H}$.

$$D_{\mathcal{H}} \quad := \quad \{\hat{h}^{v_i}\}_{i\in[d_1]} \quad \cup \quad \left\{\hat{h}^{\frac{v_i}{v_j u_k}}\right\}_{i,j\in[d_1],i\neq j,k\in[d_2]};$$

$$T_0 := \hat{h}^{1/u_0}; \quad T_1 \leftarrow_R \mathcal{H}.$$

We say that the (d_1, d_2)-q-ratio *assumption holds if for any* PPT *algorithm* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{qr}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(1^\lambda, \mathsf{par}, D_{\mathcal{G}}, D_{\mathcal{H}}, T_0) = 1] - \Pr[\mathcal{A}(1^\lambda, \mathsf{par}, D_{\mathcal{G}}, D_{\mathcal{H}}, T_1) = 1]\right|$$

is negligible in λ*.*

In this paper our focus is on constructions in prime-order groups because they are much more practical, so we will consider the q-ratio assumption on prime-order bilinear maps only. We show that this assumption is implied by the assumptions proposed by Lewko, Waters [LW12] and Attrapadung [Att14a] in Appendix C. We also show that Chen and Wee's prime order DSG construction [CW14a] (along with the new sampling algorithms we introduce) satisfies the $\mathsf{q\text{-}ratio_{dsg}}$ assumption if the underlying group satisfies the q-ratio assumption. Thus we have,

Lemma 5.8. *A dual system group with a bilinear map* $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ *that satisfies the* (d_1, d_2)-$\mathsf{q\text{-}ratio_{dsg}}$ *assumption can be instantiated in a prime-order bilinear map* $e' : \mathcal{G} \times \mathcal{H} \to \mathcal{G}_T$ *that satisfies the* (d_1, d_2)-q-ratio *and* k*-linear assumptions. Further, an element of* $\mathbb{G}$ *and* $\mathbb{H}$ *is represented using* $k + 1$ *elements of* $\mathcal{G}$ *and* $\mathcal{H}$*, respectively. (An element of* $\mathbb{G}_T$ *is represented by just one from* $\mathcal{G}_T$*).*

5.4 Encryption Scheme

In this section, we show how to obtain an encryption scheme from a pair encoding using the sampling algorithms of dual system groups. Our transformation is based on the one given by Agrawal and Chase [AC16], and is referred to as Gen-Trans. If a PES Γ_P is defined by the tuple of algorithms (Param, EncCt, EncKey, Pair) for a predicate family $P = \{P_\kappa\}_{\kappa\in\mathbb{N}^c}$, then the algorithms for $\Pi_P :=$ Gen-Trans(Γ_P) are given as follows.

- Setup(1^λ, par): First the pair encoding algorithm Param(par) is run to obtain n, and then the dual system group algorithm SampP($1^\lambda, 1^n$) is run to get PP, SP. A randomly chosen element from $\mathbb{H}$ is designated to be the master secret key MSK. Master public key MPK is set to be (PP, μ(MSK)). Further, N and κ are set to exp($\mathbb{H}$) and (N, par), respectively (where the exponent of $\mathbb{H}$ is a part of PP).

- Encrypt(MPK, x, msg): On input $x \in \mathcal{X}_\kappa$ and msg $\in \mathbb{G}_T$, EncCt(x, N) is run to obtain w_1, w_2 and polynomials $(c_1, \ldots, c_{w_3})$. For $i' \in [w_1 + w_2]^+$, draw a sample $(g_{i',0}, \ldots, g_{i',n})$ from SampG using PP. Recall that the ℓth polynomial is given by

$$\sum_{z\in[w_2]} \eta_{\ell,z}\hat{s}_z \quad + \sum_{i\in[w_1]^+, j\in[n]} \eta_{\ell,i,j} s_i b_j.$$

Set CT_i to be $g_{i,0}$ for $i \in [w_1]^+$ and $\widetilde{\mathrm{CT}}_\ell$ to be

$$\prod_{z\in[w_2]} g_{w_1+z,0}^{\eta_{\ell,z}} \quad \cdot \prod_{i\in[w_1]^+, j\in[n]} g_{i,j}^{\eta_{\ell,i,j}}$$

for $\ell \in [w_3]$. Also, let $\mathrm{CT}^\star$ = msg $\cdot$ SampGT(μ(MSK); σ) where σ denotes the coin tosses used in drawing the first sample from SampG. Output $\mathrm{CT} := (\mathrm{CT}_0, \ldots, \mathrm{CT}_{w_1}, \widetilde{\mathrm{CT}}_1, \ldots, \widetilde{\mathrm{CT}}_{w_3}, \mathrm{CT}^\star)$.

- $\mathsf{KeyGen}(\textsc{mpk}, \textsc{msk}, y)$: On input $y \in \mathcal{Y}_\kappa$, $\mathsf{EncKey}(y, N)$ is run to obtain m_1, m_2 and polynomials $(k_1, k_2, \ldots, k_{m_3})$. For $i \in [m_1 + m_2]$, draw a sample $(h_{i,0}, \ldots, h_{i,n})$ from SampH using $\textsc{pp}$. Recall the tth polynomial is given by

  $$\phi_t \alpha \quad + \sum_{z' \in [m_2]} \phi_{t,z'} \hat{r}_{z'} \quad + \sum_{i' \in [m_1], j \in [n]} \phi_{t,i',j} r_{i'} b_j.$$

  Set $\textsc{sk}_{i'}$ to be $h_{i',0}$ for $i' \in [m_1]$ and $\widetilde{\textsc{sk}}_t$ to be

  $$\textsc{msk}^{\phi_t} \quad \cdot \prod_{z' \in [m_2]} h_{m_1+z',0}^{\phi_{t,z'}} \quad \cdot \prod_{i' \in [m_1], j \in [n]} h_{i',j}^{\phi_{t,i',j}}$$

  for $t \in [m_3]$. Output $\textsc{sk} := (\textsc{sk}_1, \ldots, \textsc{sk}_{m_1}, \widetilde{\textsc{sk}}_1, \ldots, \widetilde{\textsc{sk}}_{m_3})$.

- $\mathsf{Decrypt}(\textsc{mpk}, \textsc{sk}_y, \textsc{ct}_x)$: On input $\textsc{sk}_y$ and $\textsc{ct}_x$, $\mathsf{Pair}(x, y, N)$ is run to obtain matrices $\mathbf{E}$ and $\overline{\mathbf{E}}$. Output

  $$\textsc{ct}^\star \quad \cdot \quad \left(\prod_{i \in [w_1]^+, t \in [m_3]} e(\textsc{ct}_i, \widetilde{\textsc{sk}}_t)^{E_{i,t}} \quad \cdot \prod_{\ell \in [w_3], i' \in [m_1]} e(\widetilde{\textsc{ct}}_\ell, \textsc{sk}_{i'})^{\overline{E}_{\ell,i'}} \right)^{-1}.$$

One can use the projective and associative property of DSG to show that the predicate encryption scheme defined above is correct (see [AC16] for details). We defer a proof of security for Π_P to Section 7, and conclude with the following remark.

Remark 5.9 (Size of ciphertexts and keys). Ciphertexts have $w_1 + w_3 + 1$ elements from $\mathbb{G}$ and an element from $\mathbb{G}_T$; keys have $m_1 + m_3$ elements from $\mathbb{H}$. So the size of these objects depends only on the number of non-lone variables and polynomials. Moreover, there is a one-to-one mapping between variables/polynomials and ciphertext/key elements. Thus if we can reduce the size of an encoding, we will immediately get an equivalent reduction in the size of ciphertexts or keys.

6 Transformations on Pair Encodings

In this section we present several useful transformations on pair encodings that preserve symbolic property. The first class of transformations help in reducing the size of ciphertexts and keys, and the second one provides a way to develop schemes for *dual* predicates (where the role of the two inputs to a predicate is reversed).

Compact encoding schemes. We show how pair encoding schemes can be made compact by reducing the number of ct-enc and/or key-enc polynomials and/or variables to a constant in a *generic* way. Importantly, we show that if the encoding scheme we start with satisfies the symbolic property, then so does the transformed scheme. As a result, building encryption schemes with constant-size ciphertexts or keys, for instance, becomes a very simple process.

Our first transformation converts any encoding scheme Γ' to another scheme Γ where the number of ct-enc variables is *just one*. Naturally, we need to assume a bound on the total number of ct-enc variables for this transformation to work. If $W_1 + 1$ and W_2 are bounds on the number of non-lone and lone ct-enc variables, respectively, and the number of common variables in Γ' is n, then Γ has $(W_1 + 1)n + W_2$ common variables, 1 ct-enc non-lone variable and 0 lone variables. The number of lone key-enc variables and polynomials increases by a multiplicative factor of $W_1 + 1$.

Our second transformation brings down the number of ct-enc polynomials to *just one*. Once again the transformation is fully generic, as long as there is a bound W_3 on the number of polynomials. In this case, the number of common variables increases by a multiplicative factor of $W_3 + 1$, the number of non-lone key-enc variables by a multiplicative factor of W_3, and the number of key-enc polynomials by an additive factor of $m_1 W_3^2 n$.

When the two transformations above are applied one after the other, we obtain an encoding scheme with just one non-lone variable and one polynomial in the ciphertext encoding. After augmenting the scheme as per Theorem 5.2 which adds a non-lone variable and two polynomials, we can convert the resulting encoding scheme into a predicate encryption scheme by using the generic mechanism of Section 5.4. This encryption scheme will have exactly 5 dual system's source group elements in any ciphertext, a number which would only double if the instantiation from Lemma 5.8 is used under the SXDH (1-linear) assumption.

One can also reduce the number of key-enc variables and polynomials in a manner analogous to how the corresponding quantities are reduced in the ciphertext encoding, at the cost of increasing the number of common variables and ct-enc variables and polynomials. If there is a bound on both the number of variables and polynomials in the key encoding, then one can obtain an encoding scheme with just one of each. This will result in encryption schemes with constant-size key.

Finally, we remark that one can also mix-and-match. For instance, first the number of ct-enc variables can be reduced to one, and then we can do the same for key-enc variables, resulting in a scheme with just one variable each in the ciphertext and key encodings at the cost of more polynomials in both. (This might be interesting, for example, because it produces a pair encoding of the form used in [CGW15].) Note that when the ciphertext variable reduction transformation is applied, no lone variables are left in the ciphertext encoding (the only remaining variable is a non-lone variable). Hence, the key variable reduction transformation does not affect the number of ct-enc variables.

Dual predicates. The dual predicate for a family $P'_\kappa : \mathcal{Y}_\kappa \times \mathcal{X}_\kappa \to \{0,1\}$ is given by $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ where $P_\kappa(x,y) = P'_\kappa(y,x)$ for all κ, $x \in \mathcal{X}_\kappa$, $y \in \mathcal{Y}_\kappa$. For example, CP-ABE and KP-ABE are duals of each other. In Section 6.3 we show that Attrapadung's dual scheme conversion mechanism [Att14b, Section 8.1] preserves symbolic property too.

6.1 Ciphertext Encoding Variables

In this section, the following theorem is proved.

Theorem 6.1. *If a* PES $\Gamma' = (\mathsf{Param}', \mathsf{EncCt}', \mathsf{EncKey}', \mathsf{Pair}')$ *with a bounded number of non-lone and lone* ct-enc *variables satisfies* Sym-Prop *for a predicate family* P_κ, *then there exists an encoding* Γ *with just one* ct-enc *variable (which is non-lone) that also satisfies* Sym-Prop *for* P_κ.

Proof. Suppose Γ' has bounds $W_1 + 1$ and W_2 on the number of non-lone and lone ct-enc variables, respectively. The transformation from Γ' to Γ works as follows:

- $\mathsf{Param}(\mathsf{par})$. If $\mathsf{Param}'(\mathsf{par})$ returns n, then output $W_2 + (W_1 + 1)n$. Let $\mathbf{b}$ denote the vector $(b_1, \dots, b_{W_2}, b_{0,1}, \dots, b_{0,n}, \dots, b_{W_1,1}, \dots, b_{W_1,n})$.

- $\mathsf{EncCt}(x, N)$. Run $\mathsf{EncCt}'(x, N)$ to obtain a vector $\mathbf{c}' = (c'_1, \dots, c'_{w_3})$ of polynomials, where for $\ell \in [w_3]$, c'_ℓ is given by
$$\sum_{z \in [w_2]} \eta_{\ell,z} \hat{s}'_z \quad + \sum_{i \in [w_1]^+, j \in [n]} \eta_{\ell,i,j} s'_i b'_j.$$

Define a new polynomial c_ℓ using the new common variables as

$$\sum_{z\in[w_2]} \eta_{\ell,z} s_0 b_z \quad + \sum_{i\in[w_1]^+, j\in[n]} \eta_{\ell,i,j} s_0 b_{i,j},$$

where s_0 is a new variable. Output $(c_1, \ldots, c_{w_3})$ as the new vector of polynomials in a single non-lone variable $\mathbf{s} = (s_0)$, no lone variables $\hat{\mathbf{s}} = ()$, and the common variables $\mathbf{b}$.

- $\mathsf{EncKey}(y, N)$. Run $\mathsf{EncKey}'(y, N)$ to obtain a vector $\mathbf{k}' = (k'_1, \ldots, k'_{m_3})$ of polynomials in variables $\mathbf{r}'$ and $\hat{\mathbf{r}}' = (\alpha', \hat{r}'_1, \ldots, \hat{r}'_{m_2})$. For every lone variable define $W_1 + 1$ new variables: so for α', we now have α_0, ..., α_{W_1}, and for every $\hat{r}'_z$ for $z \in [m_2]$, we have $\hat{r}_{z,0}$, ..., $\hat{r}_{z,W_1}$. Recall that k'_t is given by
$$\phi_t \alpha' \quad + \sum_{z\in[m_2]} \phi_{t,z} \hat{r}'_z \quad + \sum_{i'\in[m_1], j\in[n]} \phi_{t,i',j} r'_{i'} b'_j.$$
For each such polynomial, we define $W_1 + 1$ new polynomials as follows:
$$k_{t,i} \quad = \quad \phi_t \alpha_i \quad + \sum_{z\in[m_2]} \phi_{t,z} \hat{r}_{z,i} \quad + \sum_{i'\in[m_1], j\in[n]} \phi_{t,i',j} r'_{i'} b_{i,j}.$$
Output $(k_{1,0}, \ldots, k_{1,W_1}, \ldots, k_{m_3,0}, \ldots, k_{m_3,W_1})$ as the new set of polynomials in variables $\mathbf{r}'$, $\hat{\mathbf{r}} = (\alpha_0, \ldots, \alpha_{W_1}, \hat{r}_{1,0}, \ldots, \hat{r}_{1,W_1}, \ldots, \hat{r}_{m_2,0}, \ldots, \hat{r}_{m_2,W_1})$ and $\mathbf{b}$.

- $\mathsf{Pair}(x, y, N)$. Run $\mathsf{Pair}'(x, y, N)$ to obtain matrices $\mathbf{E}'$ and $\overline{\mathbf{E}}'$ of size $(w_1 + 1) \times m_3$ and $w_3 \times m_1$, respectively. Set $\overline{\mathbf{E}}$ to just be $\overline{\mathbf{E}}'$, but $\mathbf{E}$ is a $1 \times m_3(W_1 + 1)$ matrix with $E_{0,((t-1)(W_1+1)+i)} = E'_{i,t}$ for $i \in [w_1]^+$, $t \in [m_3]$ (rest of the entries are set to 0).

In the new encoding scheme, the monomials $\hat{s}'_z$ and $s'_i b'_j$ are being captured by introducing new common variables b_z and $b_{i,j}$, respectively. Further, the effect of multiplying s'_i with k'_t is captured by introducing $W_1 + 1$ polynomials for k'_t (where $W_1 + 1$ is the maximum number of non-lone variables in any encoding). Thus, if from the product $s'_i k'_t$, we got monomials of the form $s'_i \hat{r}'_{z'}$ and $s'_i r'_{i'} b'_j$, we now multiply s with $k_{(t-1)(W_1+1)+i}$ to get $s\hat{r}_{z',i}$ and $s r'_{i'} b_{i,j}$. Intuitively, since we are still able to preserve the *distinctness* of monomials obtained by multiplying monomials in the base encoding, we can show that correctness holds. We skip the details, which follow easily by using the matrices $\mathbf{E}$ and $\overline{\mathbf{E}}$ defined above.

Selective property. We prove the selective and co-selective properties separately. Suppose the base encoding satisfies the selective property due to the algorithms EncB', EncS' and EncR'. Then, the respective algorithms for the new encoding are as follows.

- $\mathsf{EncS}(x)$ just outputs one vector $\mathbf{s}_0$ with a single entry 1 (for s_0, the new variable).

- $\mathsf{EncB}(x)$ runs $\mathsf{EncS}'(x)$ to obtain vectors $(\mathbf{s}'_0, \ldots, \mathbf{s}'_{w_1}, \hat{\mathbf{s}}'_1, \ldots, \hat{\mathbf{s}}'_{w_2})$ and $\mathsf{EncB}'(x)$ to obtain matrices $(\mathbf{B}'_1, \ldots, \mathbf{B}'_n)$. It outputs $\hat{\mathbf{s}}'^{\top}_z$ for b_z and $\mathbf{B}'_j \mathbf{s}'^{\top}_i$ for $b_{i,j}$, where $z \in [w_2]$, $i \in [w_1]^+$ and $j \in [n]$. (It does not matter what matrices we set for the rest of the common variables.)

- $\mathsf{EncR}(x, y)$ outputs $\mathbf{a}' \mathbf{s}'^{\top}_i$ for α_i, $\hat{\mathbf{r}}'_z \mathbf{s}'^{\top}_i$ for $\hat{r}_{z,i}$, and $\mathbf{r}'_{i'}$ for $r'_{i'}$ for $z \in [m_2]$, $i \in [w_1]^+$ and $i' \in [m_1]$, where $\mathbf{s}'_0, \ldots, \mathbf{s}'_{w_1}$ are obtained by running $\mathsf{EncS}'(x)$ and rest of the vectors come from $\mathsf{EncR}'(x, y)$.

It is easy to see that all the new polynomials evaluate to zero vectors when substituted with the vectors above. Specifically, c_ℓ evaluates to

$$\sum_z \eta_{\ell,z} \hat{\mathbf{s}}'^{\top}_z \mathbf{s}_0^\top \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \mathbf{B}'_j \mathbf{s}'^{\top}_i \mathbf{s}_0^\top \quad = \quad \sum_z \eta_{\ell,z} \hat{\mathbf{s}}'^{\top}_z \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \mathbf{B}'_j \mathbf{s}'^{\top}_i,$$

which is equal to 0 by the selective symbolic property of the base scheme. Also, $k_{t,i}$ evaluates to

$$\phi_t \mathbf{a}' \mathbf{s}'^{\top}_i \quad + \quad \sum_z \phi_{t,z} \hat{\mathbf{r}}'_z \mathbf{s}'^{\top}_i \quad + \quad \sum_{i',j} \phi_{t,i',j} \mathbf{r}'_{i'} \mathbf{B}'_j \mathbf{s}'^{\top}_i,$$

which is again 0 (by selective symbolic property of the base scheme) because $\mathbf{s}'^{\top}_i$ is a common factor in every term. Also, the inner product of $\mathbf{a}' \mathbf{s}'^{\top}_0$ (the vector corresponding to α_0) with $\mathbf{s}_0$ is just $\mathbf{a}' \mathbf{s}'^{\top}_0$, which is not equal to 0.

Co-selective property. To prove the co-selective property of the new scheme, we exploit the co-selective property of the base scheme as follows. (Below we *join* matrices and vectors using the notation defined in Section 2.)

- $\mathsf{EncS}(x,y)$ runs $\mathsf{EncS}'(x,y)$ to obtain $(\mathbf{s}'_0, \ldots, \mathbf{s}'_{w_1}, \hat{\mathbf{s}}'_1, \ldots, \hat{\mathbf{s}}'_{w_2})$. It outputs $\mathbf{s}_0 = \mathbf{s}'_0 \circ \ldots \circ \mathbf{s}'_{w_1} \circ \hat{\mathbf{s}}'_1 \circ \ldots \circ \hat{\mathbf{s}}'_{w_2}$, a row vector of length $(w_1+1)d_2 + w_2 d_1$.

- $\mathsf{EncB}(y)$ runs $\mathsf{EncB}'(y)$ to get matrices $\mathbf{B}'_1, \ldots, \mathbf{B}'_n$. Set $\mathbf{B}_{i,j}$ to be $\mathbf{0} \circ \ldots \circ \mathbf{0} \circ \mathbf{B}'_j \circ \mathbf{0} \circ \ldots \circ \mathbf{0} \circ \mathbf{0}' \circ \ldots \circ \mathbf{0}'$ for $i \in [w_1]^+$, $j \in [n]$, where $\mathbf{B}'_j$ occurs at the ith position, and $\mathbf{0}$, $\mathbf{0}'$ are all-zero matrices of size $d_1 \times d_2$ and $d_1 \times d_1$, respectively (the number of $\mathbf{0}$ and $\mathbf{0}'$ matrices is w_1 and w_2, respectively). Also, $\mathbf{B}_z$ is set to $\mathbf{0} \circ \ldots \circ \mathbf{0} \circ \mathbf{0}' \circ \ldots \circ \mathbf{0}' \circ \mathbf{I} \circ \mathbf{0}' \circ \ldots \circ \mathbf{0}'$ for $z \in [w_2]$, where $\mathbf{I}$ is a d_1 dimensional identity matrix at the (w_1+1+z)th position (the number of $\mathbf{0}$ and $\mathbf{0}'$ matrices is w_1+1 and w_2-1, respectively).

- $\mathsf{EncR}(y)$ runs $\mathsf{EncR}'(y)$ to get $(\mathbf{r}'_1, \ldots, \mathbf{r}'_{m_1}, \mathbf{a}', \hat{\mathbf{r}}'_1, \ldots, \hat{\mathbf{r}}'_{m_2})$. It just outputs $\mathbf{r}'_{i'}$ for $r'_{i'}$, where $i' \in [m_1]$. But $\hat{\mathbf{r}}_{z,i}$ is set to $\mathbf{0}'' \circ \ldots \circ \mathbf{0}'' \circ \hat{\mathbf{r}}'_z \circ \mathbf{0}'' \circ \ldots \circ \mathbf{0}'' \circ \mathbf{0}^\star$, where $\mathbf{0}''$ is a row vector of size d_2 (occurring w_1 times), $\mathbf{0}^\star$ is of size $w_2 d_1$, and $\hat{\mathbf{r}}'_z$ occurs at the ith position. In a similar way, $\mathbf{a}_0, \ldots, \mathbf{a}_{w_1}$ are also set using $\mathbf{a}'$.

With matrices and vectors defined as above for various variables in the ciphertext and key encoding, one can easily show that all the polynomials evaluate to 0. Specifically, c_ℓ evaluates to

$$\sum_z \eta_{\ell,z} \mathbf{B}_z \mathbf{s}_0^\top \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \mathbf{B}_{i,j} \mathbf{s}_0^\top \quad = \quad \sum_z \eta_{\ell,z} \hat{\mathbf{s}}'^{\top}_z \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \mathbf{B}'_j \mathbf{s}'^{\top}_i,$$

and $k_{t,i}$ evaluates to

$$\begin{aligned} &\phi_t \mathbf{a}_i \quad + \quad \sum_z \phi_{t,z} \hat{\mathbf{r}}_{z,i} \quad + \quad \sum_{i',j} \phi_{t,i',j} \mathbf{r}'_{i'} \mathbf{B}_{i,j} \\ &\qquad = \mathbf{0}'' \circ \ldots \circ \mathbf{0}'' \circ \left(\phi_t \mathbf{a}' \quad + \quad \sum_z \phi_{t,z} \hat{\mathbf{r}}'_z \quad + \quad \sum_{i',j} \phi_{t,i',j} \mathbf{r}'_{i'} \mathbf{B}'_j \right) \circ \mathbf{0}'' \circ \ldots \circ \mathbf{0}'' \circ \mathbf{0}^\star, \end{aligned}$$

which are all equal to 0 due to the co-selective symbolic property of the base scheme. (In particular, note that in evaluating $k_{t,i}$, we are effectively evaluating k'_t in the ith block.) Also, $\langle \mathbf{a}_0, \mathbf{s}_0 \rangle$ is just equal to $\langle \mathbf{a}', \mathbf{s}'_0 \rangle$ because $\mathbf{a}_0$ has $\mathbf{a}'$ in the first d_2 columns and 0 everywhere else. □

Remark 6.2. Note that in the proof of co-selective symbolic property, every algorithm runs the corresponding algorithm of the underlying encoding scheme. Hence, one can use the same technique for proving selective property too. We, however, provide a separate proof to highlight that it is possible to have the dimension d_2 to be just 1. Thus, we can get selective security for the resulting encryption scheme from a potentially weaker assumption.

The same remark applies to the generic transformation for reducing the number of ct-enc polynomials in the following section.

6.2 Ciphertext Encoding Polynomials

We show how to further reduce the size of ciphertext encoding by giving a transformation that results in a scheme with just one polynomial.

Theorem 6.3. *If a* PES $\Gamma' = (\mathsf{Param}', \mathsf{EncCt}', \mathsf{EncKey}', \mathsf{Pair}')$ *with bounded number of* ct-enc *polynomials satisfies* Sym-Prop *for a predicate family* P_κ*, then there exists an encoding* Γ *with just one* ct-enc *polynomial that also satisfies* Sym-Prop *for* P_κ*. Moreover, if* Γ' *has only one variable in any ciphertext encoding which is non-lone then* Γ *preserves this property.*

Proof. We assume that PES Γ' has only one variable in the ciphertext encoding (which is non-lone). This assumption is without loss of generality if there is a bound on the number of ct-enc variables (Theorem 6.1) but even when there is no such bound, we can modify the proof below to handle this case.

Suppose s'_0 is the only ct-enc variable of Γ', then its ct-enc polynomials are given by $c'_\ell = s'_0 \sum_{j\in[n]} \eta_{j,\ell} b'_j$ for $\ell \in [w_3]$. Let W_3 be bound on the number of such polynomials in any encoding. Algorithms for the scheme Γ are now defined as follows.

- $\mathsf{Param}(\mathsf{par})$. If $\mathsf{Param}'(\mathsf{par})$ returns n, then output $(W_3+1)n$. Let $\mathbf{b}'$ denote the vector $(b'_1,\ldots,b'_n)$ and $\mathbf{b}$ denote $(b'_1,\ldots,b'_n, b_{1,1},\ldots,b_{1,W_3},\ldots,b_{n,1},\ldots,b_{n,W_3})$.

- $\mathsf{EncCt}(x,N)$. Run $\mathsf{EncCt}'(x,N)$ to obtain a vector $\mathbf{c}' = (c'_1,\ldots,c'_{w_3})$ of polynomials, where $c'_\ell = s'_0 \sum_{j\in[n]} \eta_{j,\ell} b'_j$. Define a new polynomial c using the new common variables as $s'_0 \sum_{j\in[n],\ell\in[w_3]} \eta_{j,\ell} b_{j,\ell}$. Output $\mathbf{c} = (c)$ as the new vector with a single polynomial in a single non-lone variable $\mathbf{s} = (s'_0)$ and the common variables $\mathbf{b}$.

- $\mathsf{EncKey}(y,N)$. Run $\mathsf{EncKey}'(y,N)$ to obtain a vector $\mathbf{k}' = (k'_1,\ldots,k'_{m_3})$ of polynomials in variables $\mathbf{r}' = (r'_1,\ldots,r'_{m_1})$, $\hat{\mathbf{r}}'$ and $\mathbf{b}'$. For every non-lone variable define W_3 new variables: so for $r'_{i'}$, we now have $r_{i',1}$, …, r_{i',W_3}. For every $i' \in [m_1]$, $\ell, \ell' \in [W_3]$ and $j \in [n]$, define a new polynomial $k_{i',\ell,j,\ell'}$ to be $r_{i',\ell} b_{j,\ell'}$ if $\ell \neq \ell'$, and $r_{i',\ell} b_{j,\ell} - r'_{i'} b'_j$ otherwise. Output $(\mathbf{k}', \{k_{i',\ell,j,\ell'}\})$ as the new set of polynomials in variables $\mathbf{r} = (r'_1,\ldots,r'_{m_1}, r_{1,1},\ldots,r_{1,W_3},\ldots,r_{m_1,1},\ldots,r_{m_1,W_3})$, $\hat{\mathbf{r}}'$ and $\mathbf{b}$.

- $\mathsf{Pair}(x,y,N)$. Run $\mathsf{Pair}'(x,y,N)$ to obtain matrices $\mathbf{E}'$ and $\overline{\mathbf{E}}'$ of size $1 \times m_3$ and $w_3 \times m_1$, respectively. Define $\overline{\mathbf{E}}$ of size $1 \times m_1(1+W_3)$ by setting $\overline{E}_{1,(i'-1)W_3+\ell}$ to be $\overline{E}'_{\ell,i'}$ for every $\ell \in [w_3]$, $i' \in [m_1]$ (rest of the entries are set to 0). Also, define $\mathbf{E}$ of size $1 \times (m_3 + m_1 W_3^2 n)$ by setting $\mathbf{E}_{1,t} = \mathbf{E}'_{1,t}$ for $t \in [m_3]$ and $\mathbf{E}_{1,(i',\ell,j,\ell')}$ to $-\overline{E}'_{\ell,i'} \cdot \eta_{j,\ell'}$, where by the subscript (i',ℓ,j,ℓ') we denote the column in $\mathbf{E}$ corresponding to the key-enc polynomial $k_{i',\ell,j,\ell'}$.

Although the description of the transformation may seem complex, it can be understood by considering the result of multiplying $r'_{i'}$ with c'_ℓ in the old encoding, which is $s'_0 r'_{i'} \sum_j \eta_{j,\ell} b'_j$, and how we try to mirror its effect in the new one, where we have only one polynomial c in the ciphertext encoding. We introduce several copies of both the common and the non-lone key-enc variables. In lieu of multiplying $r'_{i'}$ with c'_{ℓ}, we now multiply $r_{i',\ell}$ with c obtaining $s'_0 r_{i',\ell} \sum_{j,\ell'} \eta_{j,\ell'} b_{j,\ell'}$. But this product has terms that correspond to other $c'_{\ell'}$ in the old encoding.

In order to remove them, we introduce a new class of polynomials $\{k_{i',\ell,j,\ell'}\}$ in the key encoding. To remove $b_{j,\ell'}$ for every $\ell \neq \ell'$, we can use $r_{i',\ell} b_{j,\ell'}$ by multiplying it with $\eta_{j,\ell'}$ (and then *pairing* with s'_0), and to replace $r_{i',\ell} b_{j,\ell}$ with $r'_{i'} b'_j$, we can use $r_{i',\ell} b_{j,\ell} - r'_{i'} b'_j$ by multiplying it with $\eta_{j,\ell}$ (and then pairing with s'_0). This latter step is important because we have kept the key-enc polynomials of the old encoding intact, which are in variables $\mathbf{r}'$, $\hat{\mathbf{r}}'$ and $\mathbf{b}'$.

After removing/replacing various terms with the help of the new polynomials, the product polynomial transforms to $s'_0 r'_{i'} \sum_j \eta_{j,\ell} b'_j$, which is exactly what we had before. More formally, one can show that the new encoding scheme is correct by using the matrices $\mathbf{E}$ and $\overline{\mathbf{E}}$ defined above. We skip the details here.

Selective property. We first prove selective symbolic property. Suppose the base encoding satisfies symbolic property due to the algorithms EncB', EncS' and EncR'. Then, the respective algorithms for the new encoding are as follows.

- $\mathsf{EncS}(x)$ just outputs a vector $\mathbf{s}_0$ with a single entry 1.

- $\mathsf{EncB}(x)$ runs $\mathsf{EncS}'(x)$ to get $\mathbf{s}'_0$ and $\mathsf{EncB}'(x)$ to get $(\mathbf{B}'_1, \ldots, \mathbf{B}'_n)$. It sets $\mathbf{B}_{j,\ell}$ to be the transpose of $\mathbf{0} \circ \ldots \circ \mathbf{0} \circ (\mathbf{B}'_j {\mathbf{s}'_0}^\top)^\top \circ \mathbf{0} \circ \ldots \circ \mathbf{0}$, where $\mathbf{0}$ is a vector of size d_1 occurring $W_3 - 1$ times, and $(\mathbf{B}'_j {\mathbf{s}'_0}^\top)^\top$ appears at the ℓth position. The output is $(\mathbf{B}'_1 {\mathbf{s}'_0}^\top, \ldots, \mathbf{B}'_n {\mathbf{s}'_0}^\top, \mathbf{B}_{1,1}, \ldots, \mathbf{B}_{1,W_3}, \ldots, \mathbf{B}_{n,1}, \ldots, \mathbf{B}_{n,W_3})$.

- $\mathsf{EncR}(x, y)$ runs $\mathsf{EncS}'(x)$ to get $\mathbf{s}'_0$ and $\mathsf{EncR}'(x, y)$ to get $(\mathbf{r}'_1, \ldots, \mathbf{r}'_{m_1})$, $(\mathbf{a}', \hat{\mathbf{r}}'_1, \ldots, \hat{\mathbf{r}}'_{m_2})$. Set $\mathbf{r}_{i',\ell}$ to $\mathbf{0} \circ \ldots \circ \mathbf{0} \circ \mathbf{r}'_{i'} \circ \mathbf{0} \circ \ldots \circ \mathbf{0}$, where $\mathbf{0}$ occurs $W_3 - 1$ times like above, and $\mathbf{r}'_{i'}$ appears at the ℓth position. Output $(\mathbf{r}'_1, \ldots, \mathbf{r}'_{m_1}, \mathbf{r}_{1,1}, \ldots, \mathbf{r}_{1,W_3}, \ldots, \mathbf{r}_{m_1,1}, \ldots, \mathbf{r}_{m_1,W_3})$ and $(\mathbf{a}' {\mathbf{s}'_0}^\top, \hat{\mathbf{r}}'_1 {\mathbf{s}'_0}^\top, \ldots, \hat{\mathbf{r}}'_{m_2} {\mathbf{s}'_0}^\top)$.

When substituted with the matrices and vectors defined above, c evaluates to $\sum_{j,\ell} \eta_{j,\ell} \mathbf{B}_{j,\ell} \mathbf{s}_0^\top = \sum_\ell \sum_j \eta_{j,\ell} \mathbf{B}_{j,\ell}$. The matrix $\mathbf{B}_{j,\ell}$ is a column vector consisting of W_3 blocks with $\mathbf{B}'_j {\mathbf{s}'_0}^\top$ in the ℓth one and zero everywhere else. Thus for every ℓ, the inner-sum evaluates to zero by the selective property of the base scheme. The borrowed key-enc polynomial k'_t evaluates to

$$\phi_t \mathbf{a}' {\mathbf{s}'_0}^\top \quad + \quad \sum_z \phi_{t,z} \hat{\mathbf{r}}'_z {\mathbf{s}'_0}^\top \quad + \quad \sum_{i',j} \phi_{t,i',j} \mathbf{r}'_{i'} \mathbf{B}'_j {\mathbf{s}'_0}^\top,$$

which is again 0 (by selective property of the base scheme) because ${\mathbf{s}'_0}^\top$ is a common factor in every term. The new polynomial $r_{i',\ell} b_{j,\ell'}$ for $\ell \neq \ell'$ upon substitution gives $\mathbf{r}_{i',\ell} \mathbf{B}_{j,\ell'}$, which is zero simply because the non-zero entries in $\mathbf{r}_{i',\ell}$ and $\mathbf{B}_{j,\ell'}$ occur only in the ℓth and ℓ'th block, respectively. Finally, $r_{i',\ell} b_{j,\ell} - r'_{i'} b'_j$ gives $\mathbf{r}_{i',\ell} \mathbf{B}_{j,\ell} - \mathbf{r}'_{i'} \mathbf{B}'_j {\mathbf{s}'_0}^\top$, which is nothing but $\mathbf{r}'_{i'} \mathbf{B}'_j {\mathbf{s}'_0}^\top - \mathbf{r}'_{i'} \mathbf{B}'_j {\mathbf{s}'_0}^\top = 0$.

Co-selective property. For proving co-selective property, the algorithms are defined in the following way.

- $\mathsf{EncS}(x, y)$ just outputs the one vector $\mathbf{s}'_0$ that $\mathsf{EncS}'(x, y)$ does.

- $\mathsf{EncB}(y)$ runs $\mathsf{EncB}'(y)$ to obtain matrices $(\mathbf{B}'_1, \ldots, \mathbf{B}'_n)$. It sets $\mathbf{B}_{j,\ell}$ to be the transpose of $\mathbf{0}' \circ \ldots \circ \mathbf{0}' \circ {\mathbf{B}'_j}^\top \circ \mathbf{0}' \circ \ldots \circ \mathbf{0}'$, where $\mathbf{0}'$ is a $d_2 \times d_1$ matrix occurring $W_3 - 1$ times, and ${\mathbf{B}'_j}^\top$ appears at the ℓth position. The output is $(\mathbf{B}'_1, \ldots, \mathbf{B}'_n, \mathbf{B}_{1,1}, \ldots, \mathbf{B}_{1,W_3}, \ldots, \mathbf{B}_{n,1}, \ldots, \mathbf{B}_{n,W_3})$.

- $\mathsf{EncR}(y)$ sets $\mathbf{r}_{i',\ell}$ to be $\mathbf{0} \circ \ldots \circ \mathbf{0} \circ \mathbf{r}'_{i'} \circ \mathbf{0} \circ \ldots \circ \mathbf{0}$, where $\mathbf{0}$ occurs $W_3 - 1$ times, and $\mathbf{r}'_{i'}$, obtained from $\mathsf{EncR}'(y)$, appears at the ℓth position. The output consists of all the vectors that come from running $\mathsf{EncR}'(y)$ and the ones just defined for $i' \in [m_1]$, $\ell \in [W_3]$.

Upon substitution, c evaluates to $\sum_{j,\ell} \eta_{j,\ell} \mathbf{B}_{j,\ell} {\mathbf{s}'_0}^{\top} = \sum_{\ell} \sum_{j} \eta_{j,\ell} \mathbf{B}_{j,\ell} {\mathbf{s}'_0}^{\top}$. The matrix $\mathbf{B}_{j,\ell}$ consists of W_3 blocks joined in a row-wise fashion with $\mathbf{B}'_j$ in the ℓth block and zero everywhere else. Thus for every ℓ, the inner-sum evaluates to zero by the co-selective property of the base scheme. As far as the key-enc polynomials are concerned, $(k'_1, \dots, k'_{m_3})$ upon substitution evaluate to zero simply because they are borrowed from the base scheme, and we have not changed the vectors or matrices corresponding to the variables in them. For the new polynomial $r_{i',\ell} b_{j,\ell'}$, we can apply the same reasoning as in the selective case. Finally, $r_{i',\ell} b_{j,\ell} - r'_{i'} b'_j$ gives $\mathbf{r}_{i',\ell} \mathbf{B}_{j,\ell} - \mathbf{r}'_{i'} \mathbf{B}'_j$, which is nothing but $\mathbf{r}'_{i'} \mathbf{B}'_j - \mathbf{r}'_{i'} \mathbf{B}'_j = \mathbf{0}$. □

6.3 Dual Predicates

Given a pair encoding scheme for any predicate, an encoding scheme for the dual predicate can be designed.

Theorem 6.4. *If a* PES $\Gamma' = (\mathsf{Param}', \mathsf{EncCt}', \mathsf{EncKey}', \mathsf{Pair}')$ *satisfies* Sym-Prop *for a predicate family P_κ, then there exists an encoding Γ that satisfies* Sym-Prop *for the dual predicate of P_κ.*

Proof. The algorithms for Γ' are defined as follows:

- $\mathsf{Param}(\mathsf{par})$. If $\mathsf{Param}'(\mathsf{par})$ returns n, then output $n+1$. Let $\mathbf{b}'$ denote the vector $(b'_1, \dots, b'_n)$ and $\mathbf{b}$ denote $(b'_1, \dots, b'_n, b^\star)$.

- $\mathsf{EncCt}(x, N)$. Run $\mathsf{EncKey}'(x, N)$ to obtain a vector $\mathbf{k}'$ of polynomials in variables $\mathbf{r}'$, $\hat{\mathbf{r}}' = (\alpha', \hat{r}'_1, \dots, \hat{r}'_{m_2})$, and $\mathbf{b}'$. Output $\mathbf{c} = \mathbf{k}'$ as the new vector of polynomials, but with α' replaced by $s_0 b^\star$, where s_0 is a new variable. Hence, the polynomials in $\mathbf{c}$ are in variables $\mathbf{s} = (s_0, \mathbf{r}')$, $\hat{\mathbf{s}} = (\hat{r}'_1, \dots, \hat{r}'_{m_2})$ and $\mathbf{b}$.

- $\mathsf{EncKey}(y, N)$. Run $\mathsf{EncCt}'(y, N)$ to obtain a vector $\mathbf{c}'$ of polynomials in variables $\mathbf{s}' = (s'_0, \dots, s'_{w_1})$, $\hat{\mathbf{s}}'$ and $\mathbf{b}'$. Output $\mathbf{k} = (\mathbf{c}', \alpha - s'_0 b^\star)$ as the new vector of polynomials, where α is a new variable. Hence the polynomials in $\mathbf{k}$ are in variables $\mathbf{r} = \mathbf{s}'$, $\hat{\mathbf{r}} = (\alpha, \hat{\mathbf{s}}')$ and $\mathbf{b}$.

- $\mathsf{Pair}(x, y, N)$. Run $\mathsf{Pair}'(y, x, N)$ to obtain matrices $\mathbf{E}'$ and $\overline{\mathbf{E}}'$ of size $(w_1 + 1) \times m_3$ and $w_3 \times m_1$, respectively. Define $\mathbf{E}$ of size $(m_1 + 1) \times (w_3 + 1)$ by setting $E_{0, w_3+1} = 1$, $E_{i',\ell} = \overline{E}'_{\ell,i'}$ for $i' \in [m_1]$, $\ell \in [w_3]$, and rest of the entries to 0. $\overline{\mathbf{E}}$ is just set to be the transpose of $\mathbf{E}'$.

Correctness.

$$\begin{aligned}
\mathbf{s}\mathbf{E}\mathbf{k}^{\top} + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^{\top} &= \sum_{\substack{i' \in [m_1]^+, \\ \ell \in [w_3+1]}} s_{i'} E_{i',\ell} k_\ell \quad + \sum_{\substack{t \in [m_3], \\ i \in [w_1]^+}} c_t \overline{E}_{t,i} r_i \\
&= s_0(\alpha - s'_0 b^\star) \quad + \sum_{\substack{i' \in [m_1], \\ \ell \in [w_3]}} r'_{i'} \overline{E}'_{\ell,i'} c'_\ell \quad + \sum_{\substack{t \in [m_3], \\ i \in [w_1]^+}} k'_t E'_{i,t} s'_i \\
&= \alpha s_0 - s_0 s'_0 b^\star + (s_0 b^\star) s'_0 \quad = \quad \alpha s_0,
\end{aligned}$$

where the last but one equality is true because α' has been replaced by $s_0 b^\star$.

Selective property. We first prove selective symbolic property of Γ using the co-selective property of Γ'. Specifically,

- $\mathsf{EncS}(x)$ outputs $(\mathbf{s}_0, \mathbf{r}'_1, \dots, \mathbf{r}'_{m_1}, \hat{\mathbf{r}}'_1, \dots, \hat{\mathbf{r}}'_{m_2})$, where $\mathbf{s}_0 = (1, 0, \dots, 0)$ ($d_1 - 1$ zeroes) and rest of the vectors are obtained by running $\mathsf{EncR}'(x)$.

- $\mathsf{EncB}(x)$ runs $\mathsf{EncB}'(x)$ to obtain $\mathbf{B}'_1,\dots,\mathbf{B}'_n$ and $\mathsf{EncR}'(x)$ to get $\mathbf{a}'$ (the substitute for α'). It defines a new matrix $\mathbf{B}^\star$ of dimension $d_2 \times d_1$ with $\mathbf{a}'$ as the first column and zero everywhere else. It then outputs $(\mathbf{B}'^{\top}_1,\dots,\mathbf{B}'^{\top}_n,\mathbf{B}^\star)$.

- $\mathsf{EncR}(x,y)$ outputs $(\mathbf{s}'_0,\dots,\mathbf{s}'_{w_1},-\mathbf{s}'_0\mathbf{B}^\star,\hat{\mathbf{s}}'_1,\dots,\hat{\mathbf{s}}'_{w_2})$, where $\mathbf{B}^\star$ is obtained by running $\mathsf{EncB}(x)$ and rest of the vectors from $\mathsf{EncS}'(y,x)$.

To verify that the selective property is satisfied, observe that $\mathbf{B}^\star\mathbf{s}_0^{\top} = \mathbf{a}'^{\top}$ and hence, all $\mathsf{ct\text{-}enc}$ polynomials (which are the $\mathsf{key\text{-}enc}$ polynomials in the dual scheme) still evaluate to zero. Also, we introduced only one new polynomial $\alpha + s'_0 b^\star$ in the key encoding, and it evaluates to zero because we set $\mathbf{a}$ to be $-\mathbf{s}'_0\mathbf{B}^\star$. Further, the first entry of $\mathbf{a}$ is equal to $-\langle\mathbf{s}'_0,\mathbf{a}'\rangle$, which is not zero. Hence, $\langle\mathbf{s}_0,\mathbf{a}\rangle$ is not zero either.

Co-selective property. Before proceeding to the co-selective part of the proof, note that we cannot apply the above approach. EncR', which generates $\mathbf{a}'$, depends on both x and y in this case, so EncB with access to just y cannot set $\mathbf{B}^\star$ according to $\mathbf{a}'$ anymore. But fortunately, EncS can depend on both x, y, hence it can set $\mathbf{s}_0$ to $(\mathbf{a}',0,\dots,0)$ (so we must have d_1 at least as big as d_2). EncB just sets $\mathbf{B}^\star$ to a matrix that has the d_2-dimensional identity matrix in the first d_2 columns and zeroes everywhere else (so that $\mathbf{B}^\star\mathbf{s}_0^{\top}$ is still equal to $\mathbf{a}'^{\top}$).

Apart from how they set $\mathbf{s}_0$ and $\mathbf{B}^\star$, the three algorithms behave in the exact same way as above. In particular, even though EncR can only depend on y now, there is no problem because both EncB and EncS' need only y as input. We can easily check that all the polynomials evaluate to zero vectors of the appropriate dimension. Further, $\mathbf{a} = -\mathbf{s}'_0\mathbf{B}^\star = (-\mathbf{s}'_0,0,\dots,0)$ and we know that $\mathbf{s}_0 = (\mathbf{a}',0,\dots,0)$, so $\langle\mathbf{s}_0,\mathbf{a}\rangle \neq 0$. □

7 Security of Predicate Encryption Scheme

In this section we show that the transformation $\mathsf{Gen\text{-}Trans}$ leads to a secure encryption scheme if the underlying encoding satisfies the (enhanced) symbolic property. More formally, we have:

Theorem 7.1. *If a pair encoding scheme Γ_P satisfies (d_1,d_2)-$\mathsf{Sym\text{-}Prop}^\star$ for a predicate family P_κ, then the scheme $\mathsf{Gen\text{-}Trans}(\Gamma_P)$ defined in Section 5.4 is a fully secure predicate encryption scheme for P_κ in dual system groups under the (d_1,d_2-1)-$\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ assumption.*

When the above theorem is combined with Theorem 5.2 and Lemma 5.8, we get the following corollary:

Corollary 7.2. *If a pair encoding scheme satisfies (d_1,d_2)-$\mathsf{Sym\text{-}Prop}$ for a predicate family then there exists a fully secure predicate encryption scheme for that family in prime-order bilinear maps under the $(\max(d_1,d_2-1)+M_1+1,d_2+W_1+1)$-$\mathsf{q\text{-}ratio}$ and k-linear assumptions, where M_1 and W_1 are bounds on the number of $\mathsf{key\text{-}enc}$ and $\mathsf{ct\text{-}enc}$ non-lone variables, respectively, in the encoding.*

The rest of this section is devoted to the proof of Theorem 7.1. We follow the same general outline as in other papers that use dual system groups [CW14a, AC16, CGW15]. The design of hybrids in our proof is closer to [CW14a] and [CGW15] rather than [AC16]. In particular, our hybrid structure is simpler because, unlike [AC16], we don't add noise to individual samples in every key. However, since we have adopted the generic transformation from [AC16], the indistinguishability between several hybrids follows from that of corresponding hybrids in [AC16]. (We briefly review these hybrids and the properties they follow from below—for full proofs see [AC16].) The main novelty in our proof, and the crucial difference from [AC16], is how the form of master secret key is changed: in [AC16] relaxed perfect security is used for this purpose, but we use the symbolic property in conjunction with the $\mathsf{q\text{-}ratio_{dsg}}$ assumption.

We first define auxiliary algorithms for encryption and key generation. Below we use $g_{i,0}$ (resp. $h_{i,0}$) to denote the first element of $\mathbf{g}_i$ (resp. $\mathbf{h}_i$). Also w and m denote $w_1 + w_2$ and $m_1 + m_2$, respectively.

- $\overline{\mathsf{Encrypt}}(\textsc{pp}, x, \mathsf{msg}; (\mathbf{g}'_0, \mathbf{g}'_1, \ldots, \mathbf{g}'_w), \textsc{msk})$: This algorithm is same as $\mathsf{Encrypt}$ except that it uses $\mathbf{g}'_i \in \mathbb{G}^{n+1}$ instead of the samples $\mathbf{g}_i$ from SampG, and sets $\textsc{ct}^\star$ to $\mathsf{msg} \cdot e(g'_{0,0}, \textsc{msk})$.

- $\overline{\mathsf{KeyGen}}(\textsc{pp}, \textsc{msk}, y; (\mathbf{h}'_1, \ldots, \mathbf{h}'_m))$: This algorithm is same as KeyGen except that it uses $\mathbf{h}'_i \in \mathbb{H}^{n+1}$ instead of the samples $\mathbf{h}_i$ from SampH.

Using the algorithms described above, we define alternate forms for the ciphertext, master secret key, and secret keys.

- *Semi-functional master secret key* is defined to be $\overline{\textsc{msk}} := \textsc{msk} \cdot \tilde{h}^{\mu}$ where $\mu \leftarrow_R \mathbb{Z}_N$.

- *Semi-functional ciphertext* is given by $\overline{\mathsf{Encrypt}}(\textsc{pp}, x, m; \mathbf{G} \cdot \hat{\mathbf{G}}, \textsc{msk})$, where $\mathbf{G} \cdot \hat{\mathbf{G}}$ is defined as follows: sample $\mathbf{g}_1, \ldots, \mathbf{g}_w$ from SampG and $\hat{\mathbf{g}}_1, \ldots, \hat{\mathbf{g}}_w$ from $\overline{\mathsf{SampG}}$ (which also requires $\textsc{sp}$); set $\mathbf{G}$ and $\mathbf{G}'$ to be the vector of vectors $(\mathbf{g}_1, \ldots, \mathbf{g}_w)$ and $(\hat{\mathbf{g}}_1, \ldots, \hat{\mathbf{g}}_w)$, respectively; and denote $(\mathbf{g}_1 \cdot \hat{\mathbf{g}}_1, \ldots, \mathbf{g}_w \cdot \hat{\mathbf{g}}_w)$ by $\mathbf{G} \cdot \hat{\mathbf{G}}$.

- *Ext-semi-functional ciphertext* is given by $\overline{\mathsf{Encrypt}}(\textsc{pp}, x, m; \mathbf{G} \cdot \hat{\mathbf{G}} \cdot \hat{\mathbf{G}}', \textsc{msk})$, where $\mathbf{G}$, $\hat{\mathbf{G}}$ are as above, and $\hat{\mathbf{G}}'$ is defined to be $(\hat{\mathbf{g}}'_1, \ldots, \hat{\mathbf{g}}'_w)$, where $\hat{\mathbf{g}}'_i = (1, \hat{g}_{i,0}^{\gamma_1}, \ldots, \hat{g}_{i,0}^{\gamma_n})$ for $i \in [w]$ and $\gamma_1, \ldots, \gamma_n \leftarrow_R \mathbb{Z}_N$. (Here these $\gamma_1, \ldots, \gamma_n$ will be chosen once and used in both ciphertext and key components.)

- Table 1 lists the different types of keys we need and the inputs that should to be passed to $\overline{\mathsf{KeyGen}}$ (besides $\textsc{pp}$ and y) in order to generate them. In the table, $\mathbf{h}_1, \ldots, \mathbf{h}_m$ are samples from SampH; $\hat{\mathbf{h}}_1, \ldots, \hat{\mathbf{h}}_m$ are samples from $\overline{\mathsf{SampH}}$ (which also requires $\textsc{sp}$); and $\hat{\mathbf{h}}'_i = (1, \hat{h}_{i,0}^{\gamma_1}, \ldots, \hat{h}_{i,0}^{\gamma_n})$ for $i \in [m]$, where $\gamma_1, \ldots, \gamma_n$ are the values described above for the ext-semi-functional ciphertext.

Type of key	*Inputs to* $\overline{\mathsf{KeyGen}}$ *(besides* $\textsc{pp}$ *and* y*)*
Normal	$\textsc{msk}; (\mathbf{h}_1, \ldots, \mathbf{h}_m)$
Pseudo-normal	$\textsc{msk}; (\mathbf{h}_1 \cdot \hat{\mathbf{h}}_1, \ldots, \mathbf{h}_m \cdot \hat{\mathbf{h}}_m)$
Ext-pseudo-normal	$\textsc{msk}; (\mathbf{h}_1 \cdot \hat{\mathbf{h}}_1 \cdot \hat{\mathbf{h}}'_1, \ldots, \mathbf{h}_m \cdot \hat{\mathbf{h}}_m \cdot \hat{\mathbf{h}}'_m)$
Ext-pseudo-semi-functional	$\overline{\textsc{msk}}; (\mathbf{h}_1 \cdot \hat{\mathbf{h}}_1 \cdot \hat{\mathbf{h}}'_1, \ldots, \mathbf{h}_m \cdot \hat{\mathbf{h}}_m \cdot \hat{\mathbf{h}}'_m)$
Pseudo-semi-functional	$\overline{\textsc{msk}}; (\mathbf{h}_1 \cdot \hat{\mathbf{h}}_1, \ldots, \mathbf{h}_m \cdot \hat{\mathbf{h}}_m)$
Semi-functional	$\overline{\textsc{msk}}; (\mathbf{h}_1, \ldots, \mathbf{h}_m)$

Table 1: Six types of keys.

Let ξ denote the number of key queries made by the adversary. In Table 2, we give an outline of the proof-structure with the first column stating the various hybrids we have ($\varphi \in [\xi]$), second column describes the way in which a hybrid differs from the one in the previous row, and the third column lists the properties we need to show indistinguishability from the previous one.

To prevent the table from overflowing, we use some shorthands like ct for ciphertext, func for functional, norm for normal, msg for message, and ind for indistinguishability. Also, Hyb_0 is the game $\mathsf{IND\text{-}CPA}^b_{\mathcal{A}}(\lambda, \mathsf{par})$ which is formally defined in Section 2.1. See Appendix D for a more formal description of the hybrids.

Hybrid	*Difference from previous*	*Properties required*
Hyb_0	-	-
Hyb_1	ct semi-func	left subgroup ind
$\vdots$	$\vdots$	$\vdots$
$\mathsf{Hyb}_{2,\varphi-1,5}$	$\varphi - 1$ keys semi-func	-
$\mathsf{Hyb}_{2,\varphi,1}$	φth key pseudo-norm	right subgroup ind
$\mathsf{Hyb}_{2,\varphi,2}$	ct ext-semi-func, φth key ext-pseudo-norm	parameter hiding
$\mathsf{Hyb}_{2,\varphi,3}$	φth key ext-pseudo-semi-func	non-degeneracy, $\mathsf{Sym\text{-}Prop}^\star$, $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ assumption
$\mathsf{Hyb}_{2,\varphi,4}$	ct semi-func, φth key pseudo-semi-func	parameter-hiding
$\mathsf{Hyb}_{2,\varphi,5}$	φth key semi-func	right subgroup ind
$\vdots$	$\vdots$	$\vdots$
$\mathsf{Hyb}_{2,\xi,5}$	All keys semi-func	-
Hyb_3	ct semi-func encryption of random msg	projective, orthogonality, non-degeneracy

Table 2: An outline of the proof structure.

Our main concern here is the indistinguishability of hybrids $\mathsf{Hyb}_{2,\varphi,2}$ and $\mathsf{Hyb}_{2,\varphi,3}$ when the φth key changes from ext-pseudo-normal to ext-pseudo semi-functional, while the ciphertext stays ext-semi-functional. (Indistinguishability of the rest of the hybrids follows from [AC16] as noted earlier.) So in the rest of this section, we prove the following lemma..

Lemma 7.3. *For any* PPT *adversary* $\mathcal{A}$*, there exists a* PPT *adversary* $\mathcal{B}$ *such that the advantage of* $\mathcal{A}$ *in distinguishing* $\mathsf{Hyb}_{2,\varphi,2}$ *and* $\mathsf{Hyb}_{2,\varphi,3}$ *is at most the advantage of* $\mathcal{B}$ *in the* $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ *assumption plus some negligible quantity in the security parameter.*

7.1 Proof of Lemma 7.3

For simplicity, we prove the lemma for dual system groups where N is a prime (this suffices for Lemma 5.8 and thus Corollary 7.2). $\mathcal{B}$ gets as input a $(d_1, d_2 - 1)$ instance $(\mathrm{PP}, \mathrm{SP}, \mathsf{td}, D_{\mathbb{G}}, D_{\mathbb{H}}, T)$ of the $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ assumption, where

$$\begin{aligned} D_{\mathbb{G}} &= \{\hat{g}^{u_i}\}_{i\in[d_2-1]^+} \quad \cup \quad \left\{\hat{g}^{\frac{u_i}{u_j v_k}}\right\}_{i,j\in[d_2-1], i\neq j, k\in[d_1]}, \\ D_{\mathbb{H}} &= \{\hat{h}^{v_i}\}_{i\in[d_1]} \quad \cup \quad \left\{\hat{h}^{\frac{v_i}{v_j u_k}}\right\}_{i,j\in[d_1], i\neq j, k\in[d_2-1]}, \end{aligned}$$

for $\hat{g} \leftarrow \overline{\mathsf{SampG}_0}(\mathrm{PP},\mathrm{SP})$, $\hat{h} \leftarrow \overline{\mathsf{SampH}_0}(\mathrm{PP},\mathrm{SP})$, and T is either $\hat{h}^{1/u_0}$ or a fresh sample from $\overline{\mathsf{SampH}_0}$.

$\mathcal{B}$ first picks $\mathrm{MSK} \leftarrow_R \mathbb{H}$ and outputs $(\mathrm{PP}, \mu(\mathrm{MSK}))$ as the master public key. When $\mathcal{A}$ issues ςth key query y_ς for $\varsigma \neq \varphi$, $\mathcal{B}$ responds with

$$\begin{aligned}&\overline{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}\cdot \tilde{h}^{\nu_\varsigma}, y_\varsigma; (\mathbf{h}_1^{(\varsigma)}, \ldots, \mathbf{h}_{m^{(\varsigma)}}^{(\varsigma)})) \quad \text{if } \varsigma < \varphi \quad \text{or}\\ &\overline{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, y_\varsigma; (\mathbf{h}_1^{(\varsigma)}, \ldots, \mathbf{h}_{m^{(\varsigma)}}^{(\varsigma)})) \quad \text{if } \varsigma > \varphi,\end{aligned}$$

where $\nu_\varsigma \leftarrow_R \mathbb{Z}_N$ and $\mathbf{h}_i^{(\varsigma)} \leftarrow \mathsf{SampH}(\mathrm{PP})$ for every $\varsigma \in [\xi]$ and $i \in [m^{(\varsigma)}]$. Also, $m^{(\varsigma)}$ is the sum of m_1 and m_2 output by EncKey when given y_ς as input. (Recall that PP specifies how to sample uniformly from $\mathbb{H}$ and that SP contains $\tilde{h}$.)

7.1.1 The φth key

Running $\mathsf{Sym\text{-}Prop}^\star$ **algorithms.** When $\mathcal{A}$ issues φth key query y_φ, or simply y, there are two possibilities:

- If $\mathcal{A}$ has not yet issued a ciphertext query, $\mathcal{B}$ runs the algorithms of co-selective $\mathsf{Sym\text{-}Prop}^\star$: $\mathsf{EncB}(y)$ to obtain matrices $\mathbf{B}_1, \ldots, \mathbf{B}_n \in \mathcal{G}_N(d_1, d_2)$, and $\mathsf{EncR}(y)$ to obtain $\mathbf{r}_1, \ldots, \mathbf{r}_{m_1} \in \mathcal{G}_N(d_1)$ and $\hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2} \in \mathcal{G}_N(d_2)$.
- If $\mathcal{A}$ has already issued a ciphertext query x, $\mathcal{B}$ will instead run $\mathsf{EncR}(x, y)$ of selective $\mathsf{Sym\text{-}Prop}^\star$ to get $\mathbf{r}_1, \ldots, \mathbf{r}_{m_1}$ and $\hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2}$.

Let $\mathbf{u} = (u_0, u_1, \ldots, u_{d_2-1})$, $\mathbf{v} = (v_1, \ldots, v_{d_1})$, $\mathbf{u}^* = (1/u_0, 1/u_1, \ldots, 1/u_{d_2-1})$, and $\mathbf{v}^* = (1/v_1, \ldots, 1/v_{d_1})$. Observe that the product of $\mathbf{v}^\top$ and $\mathbf{v}^*$ gives a matrix $\mathbf{V}$ whose (i, j)th entry is v_i/v_j if $i \neq j$, otherwise it is 1. Let $\mathbf{V} = \mathbf{I} + \mathbf{V}'$ where $\mathbf{I}$ is the identity matrix of dimension d_1 and $\mathbf{V}'$ is same as $\mathbf{V}$ except it has 0s along the diagonal instead of 1s. Similarly, $\mathbf{U} = (\mathbf{u}^*)^\top \mathbf{u}$ is matrix whose (i, j)th entry is u_j/u_i if $i \neq j$, and 1 otherwise. Let $\mathbf{U} = \mathbf{I} + \mathbf{U}'$ where $\mathbf{I}$ is now the identity matrix of dimension d_2.

Form of φth key. The φth key is given by $\overline{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}', y; \mathbf{H}\cdot\hat{\mathbf{H}}\cdot\hat{\mathbf{H}}')$, where MSK' is $\mathrm{MSK}\cdot(\tilde{h}^0)$ in $\mathsf{Hyb}_{2,\varphi,2}$ and $\mathrm{MSK}\cdot(\tilde{h}^\mu)$, for $\mu \leftarrow_R \mathbb{Z}_N$, in $\mathsf{Hyb}_{2,\varphi,3}$. But we know that

$$\overline{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}', y; \mathbf{H}\cdot\hat{\mathbf{H}}\cdot\hat{\mathbf{H}}') = \overline{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, y; \mathbf{H}) \cdot \overline{\mathsf{KeyGen}}(\mathrm{PP}, \tilde{h}^\mu, y; \hat{\mathbf{H}}\cdot\hat{\mathbf{H}}'),$$

because of the way KeyGen is defined and bilinearity of e. The first component on the right hand side is the same for both the hybrids and can be generated independently using PP and MSK, so it will not be considered any further. Let $\mathrm{SK} = (\mathrm{SK}_1, \ldots, \mathrm{SK}_{m_1}, \widetilde{\mathrm{SK}}_1, \ldots, \widetilde{\mathrm{SK}}_{m_3})$ denote the output of the second component. Then we have $\mathrm{SK}_{i'} = \hat{h}_{i',0}$ for $i' \in [m_1]$ and

$$\widetilde{\mathrm{SK}}_t \quad = \quad (\tilde{h}^\mu)^{\phi_t} \quad \cdot \prod_{z'\in[m_2]} \hat{h}_{m_1+z',0}^{\phi_{t,z'}} \quad \cdot \prod_{\substack{i'\in[m_1],\\ j\in[n]}} (\hat{h}_{i',j}\cdot\hat{h}_{i',0}^{\gamma_j})^{\phi_{t,i',j}} \tag{6}$$

for $t \in [m_3]$, where $(\hat{h}_{i,0}, \ldots, \hat{h}_{i,n}) \leftarrow \overline{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP})$ for $i \in [m_1 + m_2]$ and $\gamma_1, \ldots, \gamma_n \leftarrow_R \mathbb{Z}_N$.

Simulating φth key. Recall that according to the non-degeneracy property of dual system groups, $\overline{\mathsf{SampH}_0}(\mathrm{PP}, \mathrm{SP}) \cong \tilde{h}^\delta$ if $\delta \leftarrow_R \mathbb{Z}_N$ ($\tilde{h} \neq 1_{\mathbb{H}}$). Thus, if we draw two samples $\hat{h}, h$ from $\overline{\mathsf{SampH}_0}(\mathrm{PP}, \mathrm{SP})$ then the distribution of h is statistically close to $\hat{h}^\sigma$ for a randomly chosen σ as long as $\hat{h}$ lies in the subgroup generated by $\tilde{h}$ and $\hat{h} \neq 1_{\mathbb{H}}$, which happens with overwhelming probability. More generally, the joint distribution of $(\hat{h}_{1,0},\ldots,\hat{h}_{m_1+m_2,0})$ is statistically close to $\hat{h}^{\omega_1},\ldots,\hat{h}^{\omega_{m_1+m_2}}$ for $\omega_1,\ldots,\omega_{m_1+m_2} \leftarrow_R \mathbb{Z}_N$. Also, there exists a $\beta \neq 0$ such that $\tilde{h} = \hat{h}^{\beta}$.[13] Hence, the distribution of

$$(\tilde{h}^{\mu})^{\phi_t} \quad \cdot \quad \prod_{z'} \hat{h}_{m_1+z',0}^{\phi_{t,z'}} \quad \cdot \quad \prod_{i',j} (\hat{h}_{i',0}^{\gamma_j})^{\phi_{t,i',j}}$$

part of $\widetilde{\mathrm{SK}}_t$ is statistically close to

$$\hat{h}^{\phi_t \beta\mu + \sum_{z'} \phi_{t,z'} \omega_{m_1+z'} + \sum_{i',j} \phi_{t,i',j} \omega_{i'} \gamma_j}. \tag{7}$$

We denote this part of $\widetilde{\mathrm{SK}}_t$ by $\widetilde{\mathrm{SK}}'_t$.

$\mathcal{B}$ implicitly sets $\omega_{i'}$ to $\langle \mathbf{r}_{i'}, \mathbf{v}\rangle$ for $i' \in [m_1]$; $\omega_{m_1+z'}$ to $\langle \hat{\mathbf{r}}_{z'}, \mathbf{u}^*\rangle + \varepsilon_{z'} v_1$ for $z' \in [m_2]$[14], where $\varepsilon_1,\ldots,\varepsilon_{m_2} \leftarrow_R \mathbb{Z}_N$; and γ_j to $\langle \mathbf{v}^*\mathbf{B}_j, \mathbf{u}^*\rangle + \epsilon_j$ for $j \in [n]$, where $\epsilon_1,\ldots,\epsilon_n \leftarrow_R \mathbb{Z}_N$. (If the φth key request is made after the challenge ciphertext phase then $\gamma_1,\ldots,\gamma_n$ have already been set to the same value.) It is easy to see that the implicit assignment to variables $\omega_{m_1+1},\ldots,\omega_{m_1+m_2},\gamma_1,\ldots,\gamma_n$ are independently and uniformly distributed. Also, Sym-Prop* guarantees that the vectors $\mathbf{r}_1,\ldots,\mathbf{r}_{m_1}$ are independent of each other. Hence, $\langle \mathbf{r}_1, \mathbf{v}\rangle,\ldots,\langle \mathbf{r}_{m_1}, \mathbf{v}\rangle$ are also uniformly distributed, independent of other variables.

The key components $\mathrm{SK}_1,\ldots,\mathrm{SK}_{m_1}$, that are just $\hat{h}^{\omega_1},\ldots,\hat{h}^{\omega_{m_1}}$, can be easily generated using the terms $\hat{h}^{v_1},\ldots,\hat{h}^{v_{d_1}}$ terms from the assumption. In order to generate $\widetilde{\mathrm{SK}}_t$, we need to generate both $\widetilde{\mathrm{SK}}'_t$ and the product of $\hat{h}_{i',j}$ terms in (6). For the first part, observe that the power of $\hat{h}$ in (7) is being implicitly set to

$$\phi_t\beta\mu \quad + \quad \sum_{z'} \phi_{t,z'} \left[\left\langle \hat{\mathbf{r}}_{z'}, \mathbf{u}^*\right\rangle + \varepsilon_{z'} v_1\right] \quad + \quad \sum_{i',j} \phi_{t,i',j} \left\langle \mathbf{r}_{i'}, \mathbf{v}\right\rangle \left[\left\langle \mathbf{v}^*\mathbf{B}_j, \mathbf{u}^*\right\rangle + \epsilon_j\right]$$

$$\begin{aligned} = \quad & \phi_t\beta\mu \quad + \quad \sum_{z'} \phi_{t,z'} \left[\left\langle \hat{\mathbf{r}}_{z'}, \mathbf{u}^*\right\rangle + \varepsilon_{z'} v_1\right] \\ & \qquad + \quad \sum_{i',j} \phi_{t,i',j} \left\langle \mathbf{r}_{i'}\mathbf{B}_j, \mathbf{u}^*\right\rangle + \phi_{t,i',j} \left[\left\langle \mathbf{r}_{i'}\mathbf{V}'\mathbf{B}_j, \mathbf{u}^*\right\rangle + \left\langle \mathbf{r}_{i'}, \mathbf{v}\right\rangle \epsilon_j\right] \end{aligned}$$

$$\begin{aligned} = \quad & \boxed{\phi_t\beta\mu} \quad + \quad \boxed{\left\langle \sum_{z'} \phi_{t,z'} \hat{\mathbf{r}}_{z'} \quad + \quad \sum_{i',j} \phi_{t,i',j} \mathbf{r}_{i'}\mathbf{B}_j \quad , \quad \mathbf{u}^* \right\rangle} \\ & \qquad + \quad \boxed{\sum_{z'} \phi_{t,z'} \varepsilon_{z'} v_1} \quad + \quad \boxed{\sum_{i',j} \phi_{t,i',j} \left[\left\langle \mathbf{r}_{i'}\mathbf{V}'\mathbf{B}_j, \mathbf{u}^*\right\rangle + \left\langle \mathbf{r}_{i'}, \mathbf{v}\right\rangle \epsilon_j\right]}, \end{aligned}$$

where the second equality follows because $\mathbf{v}^\top \mathbf{v}^* = \mathbf{V} = \mathbf{I} + \mathbf{V}'$.

The third component in the final expression can be generated from the term $\hat{h}^{v_1}$ in the assumption. The fourth component is a sum over $\phi_{t,i',j}\left[\left\langle \mathbf{r}_{i'}\mathbf{V}'\mathbf{B}_j, \mathbf{u}^*\right\rangle + \left\langle \mathbf{r}_{i'}, \mathbf{v}\right\rangle \epsilon_j\right]$. In this, $\phi_{t,i',j} \left\langle \mathbf{r}_{i'}, \mathbf{v}\right\rangle \epsilon_j$ can be generated in the same way as previous key components, using $\hat{h}^{v_1},\ldots,\hat{h}^{v_{d_1}}$. We have to be more careful with the remaining part because $1/u_0$ is not available in any of the $\mathbb{H}$ terms of the assumption. So we must show that the first element of the vector $\mathbf{p} := \sum_{i',j} \phi_{t,i',j} \mathbf{r}_{i'}\mathbf{V}'\mathbf{B}_j$ is 0. This follows from Lemma 5.5 because $\sum_{i',j} \phi_{t,i',j} \mathbf{b}_j^\top \mathbf{r}_{i'}$ is a diagonal matrix due to Sym-Prop* ($\mathbf{b}_j$ is the first column of $\mathbf{B}_j$) and $\mathbf{V}'$ has zeroes in the diagonal. Now $\hat{h}^{\langle \mathbf{p}, \mathbf{u}^*\rangle}$ can be generated using the $\hat{h}^{v_i/(v_j u_k)}$ terms from the assumption.

[13] Recall that for simplicity we have restricted ourselves to dual system groups where N is a prime.

[14] The choice of v_1 is arbitrary. Any other element of $\mathbf{v}$ is equally good.

Suppose we have generated the third and fourth components of $\widetilde{\text{SK}}'_t$, as described above. The second component is an inner product, the first vector of which is $(-\phi_t, 0, \ldots, 0)$ due to $\mathsf{Sym\text{-}Prop}^\star$.[15] Hence, the inner product evaluates to $-\phi_t/u_0$. Putting the first and second components together, we have $\phi_t(\beta\mu - 1/u_0)$. If $\phi_t = 0$ then the simulation of $\widetilde{\text{SK}}'_t$ is complete. Otherwise, we compute this part as $T^{-\phi_t}$ where T is the challenge term in the assumption, which is either $\hat{h}^{1/u_0}$ or a fresh sample from $\overline{\mathsf{SampH}}_0$. Recall that $\mu = 0$ in $\mathsf{Hyb}_{2,\varphi,2}$ and μ is random in $\mathsf{Hyb}_{2,\varphi,3}$. In the former hybrid, $\hat{h}^{\phi_t(\beta\mu-1/u_0)}$ becomes $\hat{h}^{-\phi_t/u_0}$, which is exactly what we generate when $T = \hat{h}^{1/u_0}$. In the latter hybrid, $\hat{h}^{\phi_t(\beta\mu-1/u_0)}$ is a random element in the subgroup generated by $\tilde{h}$, and so is $T^{-\phi_t}$ when T is a fresh sample from $\overline{\mathsf{SampH}}_0$ (with overwhelming probability).

In order to complete the simulation of $\widetilde{\text{SK}}_t$, its second part which is a product of $\hat{h}_{i',j}$ terms still needs to be generated. This can be done using the $\mathsf{ExtendH}$ algorithm with inputs PP, SP, td and $\hat{h}^{\langle \mathbf{r}_{i'}, \mathbf{v} \rangle}$. The first three come from the assumption and we have already seen how to generate the last one (it is the implicit representation of the i'th key component $\text{SK}_{i'}$). Let $\mathsf{ExtendH}_{i',j}$ denote the jth output of $\mathsf{ExtendH}$ when given these inputs. Now, $\widetilde{\text{SK}}_t$ is generated as:

$$\widetilde{\text{SK}}'_t \quad \cdot \quad \prod_{i',j} (\mathsf{ExtendH}_{i',j})^{\phi_{t,i',j}},$$

7.1.2 The challenge ciphertext

Running $\mathsf{Sym\text{-}Prop}^\star$ **algorithms.** When $\mathcal{A}$ sends a challenge x and two messages m_0, m_1, there are two possibilities:

- If $\mathcal{A}$ has not yet made the φth key query, $\mathcal{B}$ runs the algorithms of selective $\mathsf{Sym\text{-}Prop}^\star$: $\mathsf{EncB}(x)$ to obtain matrices $\mathbf{B}_1, \ldots, \mathbf{B}_n \in \mathcal{G}_N(d_1, d_2)$, and $\mathsf{EncS}(x)$ to obtain $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{w_1} \in \mathcal{G}_N(d_2)$ and $\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2} \in \mathcal{G}_N(d_1)$.
- If $\mathcal{A}$ has already issued the φth key query y, $\mathcal{B}$ will instead run $\mathsf{EncS}(x, y)$ of co-selective $\mathsf{Sym\text{-}Prop}^\star$ to get $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{w_1}$ and $\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2}$.

The form of ciphertext. The challenge ciphertext is given by $\overline{\mathsf{Encrypt}}(\text{PP}, x, m_b; \mathbf{G} \cdot \hat{\mathbf{G}} \cdot \hat{\mathbf{G}}', \text{MSK})$. But we know that

$$\overline{\mathsf{Encrypt}}(\text{PP}, x, m_b; \mathbf{G} \cdot \hat{\mathbf{G}} \cdot \hat{\mathbf{G}}', \text{MSK}) = \overline{\mathsf{Encrypt}}(\text{PP}, x, m_b; \mathbf{G}, \text{MSK}) \cdot \overline{\mathsf{Encrypt}}(\text{PP}, x, 1; \hat{\mathbf{G}} \cdot \hat{\mathbf{G}}', \text{MSK}).$$

The first component on the right hand side can be generated independently using only PP and MSK, so it will not be considered any further. Let $\text{CT} = (\text{CT}_0, \ldots, \text{CT}_{w_1}, \widetilde{\text{CT}}_1, \ldots, \widetilde{\text{CT}}_{w_3}, \text{CT}^\star)$ denote the output of the second component. Then we have $\text{CT}_i = \hat{g}_{i,0}$ for $i \in [w_1]^+$ and

$$\widetilde{\text{CT}}_\ell \quad = \quad \prod_{z \in [w_2]} \hat{g}_{w_1+z,0}^{\eta_{\ell,z}} \quad \cdot \quad \prod_{\substack{i \in [w_1]^+, \\ j \in [n]}} (\hat{g}_{i,j} \cdot \hat{g}_{i,0}^{\gamma_j})^{\eta_{\ell,i,j}} \tag{8}$$

for $\ell \in [w_3]$, where $(\hat{g}_{i',0}, \ldots, \hat{g}_{i',n}) \leftarrow \overline{\mathsf{SampG}}(\text{PP}, \text{SP})$ for $i' \in [w_1 + w_2]^+$ (and $\gamma_1, \ldots, \gamma_n$ are same as before). Also, $\text{CT}^\star$ is given by $e(\hat{g}_{0,0}, \text{MSK})$.

[15] To see this, note that we have removed the term corresponding to $\phi_t \alpha$ in the polynomial k_t. Thus, the resulting equation should be equal to $-\phi_t(1, 0, \ldots, 0) = (-\phi_t, 0, \ldots, 0)$.

Simulating the ciphertext. The non-degeneracy property of dual system groups gives us that $\exists$ $\tilde{g} \in \mathbb{G}$ s.t. $\tilde{g} \neq 1_{\mathbb{G}}$ and $\overline{\mathsf{SampG}}_0(\textsc{pp}, \textsc{sp}) \cong \tilde{g}^{\alpha}$, where $\alpha \leftarrow_R \mathbb{Z}_N$. So for a $\hat{g} \leftarrow \overline{\mathsf{SampG}}_0(\textsc{pp}, \textsc{sp})$, the joint distribution of $(\hat{g}_{0,0}, \ldots, \hat{g}_{w_1+w_2,0})$ is statistically close to $(\hat{g}^{\delta_0}, \ldots, \hat{g}^{\delta_{w_1+w_2}})$ if $\delta_0, \ldots, \delta_{w_1+w_2} \leftarrow_R \mathbb{Z}_N$ (as long as $\hat{g} \neq 1_{\mathbb{G}}$ which happens with overwhelming probability). Therefore, the distribution of

$$\prod_z \hat{g}_{w_1+z,0}^{\eta_{\ell,z}} \quad \cdot \quad \prod_{i,j} (\hat{g}_{i,0}^{\gamma_j})^{\eta_{\ell,i,j}}$$

part of $\widetilde{\textsc{ct}}_\ell$ is statistically close to

$$\hat{g}^{\sum_z \eta_{\ell,z} \delta_{w_1+z} + \sum_{i,j} \eta_{\ell,i,j} \delta_i \gamma_j}. \tag{9}$$

$\mathcal{B}$ implicitly sets δ_i to $\langle \mathbf{u}, \mathbf{s}_i \rangle$ for $i \in [w_1]^+$; δ_{w_1+z} to $\langle \mathbf{v}^*, \hat{\mathbf{s}}_z \rangle + \varrho_z u_0$, where $\varrho_z \leftarrow_R \mathbb{Z}_N$, for $z \in [w_2]$; and γ_j to $\left\langle \mathbf{v}^* \mathbf{B}_j, \mathbf{u}^* \right\rangle + \epsilon_j$ for $j \in [n]$, where $\epsilon_1, \ldots, \epsilon_n \leftarrow_R \mathbb{Z}_N$. (If the φth key request is made before the challenge ciphertext phase then $\gamma_1, \ldots, \gamma_n$ have already been set to the same value.) Clearly, the implicit assignments to $\delta_{w_1+1}, \ldots, \delta_{w_1+w_2}, \gamma_1, \ldots, \gamma_n$ are independently and uniformly distributed. And moreover, since $\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}$ are independent vectors, $\langle \mathbf{u}, \mathbf{s}_0 \rangle, \ldots, \left\langle \mathbf{u}, \mathbf{s}_{w_1} \right\rangle$ are also uniformly distributed, independent of others.

Once again, the first $w_1 + 1$ ciphertext components can be easily generated because they are just equal to $\hat{g}_{i,0}$ or $\hat{g}^{\delta_i}$ for $i \in [w_1]^+$ using the terms $\hat{g}^{u_0}, \ldots, \hat{g}^{u_{d_2-1}}$ from the assumption. For the remaining components $\widetilde{\textsc{ct}}_1, \ldots, \widetilde{\textsc{ct}}_{w_3}$, the expression in the power of $\hat{g}$ in (9) is being implicitly set to

$$\sum_z \eta_{\ell,z} \left[\langle \mathbf{v}^*, \hat{\mathbf{s}}_z \rangle + \varrho_z u_0 \right] \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \left[\langle \mathbf{v}^* \mathbf{B}_j, \mathbf{u}^* \rangle + \epsilon_j \right] \langle \mathbf{u}, \mathbf{s}_i \rangle$$

$$= \quad \sum_z \eta_{\ell,z} \left[\langle \mathbf{v}^*, \hat{\mathbf{s}}_z \rangle + \varrho_z u_0 \right] \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \left\langle \mathbf{v}^*, \mathbf{s}_i \mathbf{B}_j^{\mathsf{T}} \right\rangle \; + \; \eta_{\ell,i,j} \left[\left\langle \mathbf{v}^*, \mathbf{s}_i {\mathbf{U}'}^{\mathsf{T}} \mathbf{B}_j^{\mathsf{T}} \right\rangle + \epsilon_j \langle \mathbf{u}, \mathbf{s}_i \rangle \right]$$

$$= \boxed{\left\langle \mathbf{v}^* \quad , \quad \sum_z \eta_{\ell,z} \hat{\mathbf{s}}_z \quad + \quad \sum_{i,j} \eta_{\ell,i,j} \mathbf{s}_i \mathbf{B}_j^{\mathsf{T}} \right\rangle}$$

$$+ \quad \boxed{\sum_z \eta_{\ell,z} \varrho_z u_0} \quad + \quad \boxed{\sum_{i,j} \eta_{\ell,i,j} \left[\left\langle \mathbf{v}^*, \mathbf{s}_i {\mathbf{U}'}^{\mathsf{T}} \mathbf{B}_j^{\mathsf{T}} \right\rangle + \epsilon_j \langle \mathbf{u}, \mathbf{s}_i \rangle \right]},$$

where the second equality follows because $(\mathbf{u}^*)^{\mathsf{T}} \mathbf{u} = \mathbf{U} = \mathbf{I} + \mathbf{U}'$.

The first term goes to 0 due to the symbolic property. The second can be easily computed from the $\hat{g}^{u_0}$ term of the assumption. In the third, $\langle \mathbf{u}, \mathbf{s}_i \rangle$ is generated in the same way as previous ciphertext components. The remaining term has $\mathbf{U}'$ but u_0 is not available in the numerator or denominator of $\hat{g}^{u_i/(u_j v_k)}$ terms in the assumption. Let $s_{i,1}$ be the first element of $\mathbf{s}_i$ and $\mathbf{b}_j$ be the first column of $\mathbf{B}_j$. Then recall that $\sum_{i,j} \eta_{\ell,i,j} \mathbf{s}_i^{\mathsf{T}} \mathbf{b}_j$ and $\sum_{i,j} \eta_{\ell,i,j} s_{i,1} \mathbf{B}_j$ are matrices with non-zero values in the first row and first column only, respectively, due to $\mathsf{Sym\text{-}Prop}^{\star}$. Thus the vector $\mathbf{p}^{\mathsf{T}} := \sum_{i,j} \eta_{\ell,i,j} \mathbf{B}_j \mathbf{U}' \mathbf{s}_i^{\mathsf{T}}$ does not depend on the first row and column of $\mathbf{U}'$, which are the only places u_0 appears, due to Lemma 5.3 and 5.4. Therefore, $\hat{g}^{\langle \mathbf{v}^*, \mathbf{p} \rangle}$ can be generated from the $\hat{g}^{u_i/(u_j v_k)}$ terms of the assumption.

As far as the product of $\hat{g}_{i,j}$ terms in (8) is concerned, we handle them in the same manner as the corresponding key components. $\mathcal{B}$ uses $\mathsf{ExtendG}$ with inputs $\textsc{pp}$, $\textsc{sp}$, td, and $\hat{g}^{\langle \mathbf{u}, \mathbf{s}_i \rangle}$ to get $\hat{g}_{i,1}, \ldots, \hat{g}_{i,n}$ for $i \in [w_1]^+$.

Finally, $\mathcal{B}$ needs to simulate $\textsc{ct}^{\star}$. Its distribution is statistically close to $e(\hat{g}^{\delta_0}, \textsc{msk})$, which is implicitly set to $e(\hat{g}^{\langle \mathbf{u}, \mathbf{s}_0 \rangle}, \textsc{msk})$, and can be generated by pairing $\textsc{ct}_0$ with $\textsc{msk}$.

References

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

[AHY15] Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 575–601. Springer, Heidelberg, November / December 2015.

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

[ALdP11] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 90–108. Springer, Heidelberg, March 2011.

[Att14a] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014.

[Att14b] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully-secure functional encryption for regular languages, and more. Cryptology ePrint Archive, Report 2014/428, 2014. http://eprint.iacr.org/2014/428.

[Att15] Nuttapong Attrapadung. Dual system encryption framework in prime-order groups. Cryptology ePrint Archive, Report 2015/390, 2015. http://eprint.iacr.org/2015/390.

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

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

[Bei11] Amos Beimel. Secret-sharing schemes: A survey. In YeowMeng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46. Springer Berlin Heidelberg, 2011.

[Boy13] Xavier Boyen. Attribute-based functional encryption on lattices. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 122–142. Springer, Heidelberg, March 2013.

[BRS13] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 461–478. Springer, Heidelberg, August 2013.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007.

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 535–554. Springer, Heidelberg, February 2007.

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

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

[CW14a] Jie Chen and Hoeteck Wee. Dual system groups and its applications — compact HIBE and more. Cryptology ePrint Archive, Report 2014/265, 2014. http://eprint.iacr.org/2014/265.

[CW14b] Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In Michel Abdalla and Roberto De Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 277–297. Springer, Heidelberg, September 2014.

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

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGHZ16] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional encryption without obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 480–511. Springer, Heidelberg, January 2016.

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

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

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

[HHH$^+$14] Gottfried Herold, Julia Hesse, Dennis Hofheinz, Carla Ràfols, and Andy Rupp. Polynomial spaces: A new framework for composite-to-prime-order transformations. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 261–279. Springer, Heidelberg, August 2014.

[KL15] Lucas Kowalczyk and Allison Bishop Lewko. Bilinear entropy expansion from the decisional linear assumption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 524–541. Springer, Heidelberg, August 2015.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, April 2008.

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

[LW11a] Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 568–588. Springer, Heidelberg, May 2011.

[LW11b] Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Heidelberg, May 2011.

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

[OSW07] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 07*, pages 195–203. ACM Press, October 2007.

[OT12] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Heidelberg, December 2012.

[RW13] Yannis Rouselakis and Brent Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13*, pages 463–474. ACM Press, November 2013.

[SSW09] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 457–473. Springer, Heidelberg, March 2009.

[SSW12] Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 199–217. Springer, Heidelberg, August 2012.

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

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

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

[Wat15] Brent Waters. A punctured programming approach to adaptively secure functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 678–697. Springer, Heidelberg, August 2015.

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

[YAHK14] Shota Yamada, Nuttapong Attrapadung, Goichiro Hanaoka, and Noboru Kunihiro. A framework and compact constructions for non-monotonic attribute-based encryption. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 275–292. Springer, Heidelberg, March 2014.

A Pair Encoding Schemes: Prior Formulation

We describe here the formulation of pair encoding schemes due to Agrawal and Chase [AC16], which is same as the one given by Attrapadung [Att15].

- $\mathsf{Param}(\mathsf{par}) \to n$. When given par as input, Param outputs an $n \in \mathbb{N}$ that specifies the number of common variables, which is denoted by $\mathbf{b} := (b_1, b_2, \ldots, b_n)$.

- $\mathsf{EncCt}(x, N) \to (\mathbf{c}; w_2)$. On input $N \in \mathbb{N}$ and $x \in \mathcal{X}_{(N,\mathsf{par})}$, EncCt outputs a vector of w_1 polynomials $\mathbf{c} := (c_1, c_2, \ldots, c_{w_1})$ and a non-negative integer w_2, where for $\ell \in [w_1]$,

$$c_\ell \;=\; \zeta_\ell s \;+\; \sum_{i \in [w_2]} \eta_{\ell,i} s_i \;+\; \sum_{j \in [n]} \theta_{\ell,j} s b_j \;+ \sum_{\substack{i \in [w_2],\\ j \in [n]}} \vartheta_{\ell,i,j} s_i b_j,$$

  and $\zeta_\ell, \eta_{\ell,i}, \theta_{\ell,j}, \vartheta_{\ell,i,j} \in \mathbb{Z}_N$.

  Further, for any $i \in [w_2]$, if $\vartheta_{\ell,i,j} \neq 0$ for some ℓ and j, then $s_i \in \mathbf{c}$[16]. In simple words, if s_i *appears with* a b_j in any polynomial, then it must be *explicitly given out*. It is also required that $s \in \mathbf{c}$.

[16] We are viewing the vector $\mathbf{c}$ of polynomials as a set here.

- $\mathsf{EncKey}(y, N) \to (\mathbf{k}; m_2)$. On input $N \in \mathbb{N}$ and $y \in \mathcal{Y}_{(N,\mathsf{par})}$, EncKey outputs a vector of m_1 polynomials $\mathbf{k} := (k_1, k_2, \ldots, k_{m_1})$, where for $t \in [m_1]$,

$$k_t \quad = \quad \tau_t \alpha \quad + \sum_{i' \in [m_2]} v_{t,i'} r_{i'} \quad + \sum_{\substack{i' \in [m_2], \\ j \in [n]}} \phi_{t,i',j} r_{i'} b_j,$$

  and $\tau_t, v_{t,i'}, \phi_{t,i',j} \in \mathbb{Z}_N$.

  Once again, for any $i' \in [m_2]$, if $\phi_{t,i',j} \neq 0$ for any t and j, then $r_{i'} \in \mathbf{k}$.

- $\mathsf{Pair}(x, y, N) \to \mathbf{E}$. On input N, and both x and y, Pair outputs a matrix $\mathbf{E}$ of size $m_1 \times w_1$ with entries in $\mathbb{Z}_N$, such that for any ℓ and t, if $\phi_{t,i',j'}$ and (at least) one of $\theta_{\ell,j}$ or $\vartheta_{\ell,i,j}$ are non-zero (for some $i = [w_2]$, $i' \in [m_2]$, and $j, j' \in [n]$), then $E_{t,\ell} = 0$.

Correctness. A pair encoding scheme is correct if for every $\kappa = (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $P_\kappa(x, y) = 1$, the following holds symbolically

$$\mathbf{k}\mathbf{E}\mathbf{c}^\top = \sum_{\substack{t \in [m_1], \\ \ell \in [w_1]}} E_{t,\ell} k_t c_\ell = \alpha s.$$

An alternate formulation. We now describe why it is sufficient to work with a restricted form of pair encodings without losing any generality. More formally, we show that any pair encoding of the form above can be converted into one of a special form without affecting any known security property like perfect, computational [Att15] or relaxed perfect [AC16].

Let $V = \{s, s_1, \ldots, s_{w_2}, \alpha, r_1, \ldots, r_{m_2}\}$ be the set of all variables in a pair encoding. We say that a variable $v \in V$ is *non-lone* if it appears with a b_j in some polynomial (otherwise, it is lone). Specifically, s is non-lone if $\theta_{\ell,j} \neq 0$ for some ℓ, j; for $i \in [w_2]$, s_i is non-lone if $\vartheta_{\ell,i,j} \neq 0$ for some ℓ, j; and, for $i' \in [m_2]$, $r_{i'}$ is non-lone if $\phi_{t,i',j} \neq 0$ for some t, j. We know that α is always lone because terms of the form αb_j are not allowed.

If there is a polynomial of the form $p = \sigma v$ ($\sigma \in \mathbb{Z}_N$, $v \in V$) in the encoding, then v can be removed from any other polynomial p' by replacing p' with $\sigma p' - \sigma' p$, where σ' is the coefficient of v in p'. It is easy to see that by modifying $\mathbf{E}$ appropriately, correctness still holds. Security properties are not affected because an adversary can do such replacements himself, even if the polynomials are in the exponent of some group element like in Attrapadung's computational property.

Once the above transformation is applied, note that a non-lone variable v does not appear with other terms in any polynomial, because by definition of pair encodings, there is always a polynomial of the form $p = v$. The same holds for the variable s, so from now on we treat it as a non-lone variable.

If a polynomial of the form $\hat{p} = \sigma \hat{v}$ is available for a lone variable $\hat{v} \neq \alpha$, then we call $\hat{v}$ a *frivolous* variable. As discussed above, all other occurrences of $\hat{v}$ can be removed from the encoding. But, since $\hat{v}$ does not occur with a b_j, $\hat{p}$ is the only polynomial left that has $\hat{v}$ in any form. Now this polynomial can also be removed without affecting correctness—just set all entries of $\mathbf{E}$ that pair $\hat{p}$ with another polynomial to 0 (security properties are obviously preserved).

Call the polynomials that just have a single (non-lone) variable *simple*, and any other polynomial *non-simple*. Once the above transformations have been applied, a non-simple polynomial can only be one of two types: either it is a sum of lone variables only (type-I) (but it may also include polynomials of the form $\sigma\alpha$), or it additionally has terms of the form $s b_j$, $s_i b_j$ or $r_{i'} b_j$ (type-II). In the following, we will get rid of all type-I polynomials.

Let $p = \sigma\hat{v} + q$ be a type-I polynomial where $\hat{v} \neq \alpha$ and q is a sum of lone variables. Replace $\hat{v}$ by $\hat{v} - \sigma^{-1}q$ in all the polynomials including p. This does not violate correctness because $\mathbf{kEc}^\top$ does not have a $\hat{v}$ term. Further, $\hat{v} - \sigma^{-1}q$ is an independent and uniformly distributed random variable like $\hat{v}$, and so all the security properties are preserved. Also, note that a type-II polynomial cannot become type-I due to this replacement.

$\hat{v}$ is now a frivolous variable. Therefore, all occurrences of $\hat{v}$ in other polynomials can be done away with, and the modified polynomial p (which is now just $\sigma\hat{v}$) can also be removed. Thus, we removed at least one type-I polynomial and a lone variable.

We can keep repeating the process above as long as we have a type-I polynomial of the form $\sigma_1\hat{v}_1 + \sigma_2\hat{v}_2 + \ldots$ where at least one of the variables is not α. Since we get rid of a lone variable each time, there could be at most $w_2 + m_2$ iterations. The only type-I polynomials that may be left at the end are of the form $p^\star = \sigma\alpha$, but we claim that this is not possible. Note that all the transformations we have applied to the original encoding preserve all the security properties, so the resulting scheme with $p^\star$ must also be secure. However, one can easily distinguish between α being 0 or chosen at random given $p^\star$ in plain or in exponent—thus, making the modified encoding insecure. We thus conclude that no polynomials of type-I are left.

In light of the above discussion, we can simplify the description of the vector of polynomials **c** and **k** output by EncCt and EncKey, respectively. Several of these polynomials are simple: they consist of just a non-lone variable. Excluding them, suppose we are left with $\overline{w}_3$ and $\overline{m}_3$ number of (type-II) non-simple polynomials in **c** and **k** respectively. Let $\overline{w}_1 + 1$ and $\overline{m}_1$ be the number of non-lone variables in the encoding of x and y respectively. Similarly, let $\overline{w}_2$ and $\overline{m}_2$ be the number of lone variables. Then, the non-simple polynomials are given by

$$c_\ell \quad = \quad \sum_{z\in[\overline{w}_2]} \eta_{\ell,z}\hat{s}_z \quad + \sum_{i\in[\overline{w}_1]^+, j\in[n]} \eta_{\ell,i,j}s_i b_j,$$

and

$$k_t \quad = \quad \phi_t\alpha \quad + \sum_{z'\in[\overline{m}_2]} \phi_{t,z'}\hat{r}_{z'} \quad + \sum_{i'\in[\overline{m}_1], j\in[n]} \phi_{t,i',j} r_{i'} b_j,$$

for $\ell \in [\overline{w}_3]$ and $t \in [\overline{m}_3]$, where we distinguish the lone variables from non-lone by using a hat symbol, and use s_0 to represent the variable s. Note that in any c_ℓ (resp. k_t), not all $\eta_{\ell,i,j}$ (resp. $\phi_{t,i',j}$) can be zero.

Finally, we turn our attention to the matrix **E** output by the Pair algorithm. We know that two polynomials, one with $s_i b_j$ and other with $r_{i'} b_{j'}$, cannot be multiplied with each other. In other words, the non-simple polynomials we have defined above can never be paired. Further, pairing simple polynomials, like an s_i with an $r_{i'}$, is not useful either, because there is no other product that can generate the monomial $s_i r_{i'}$. Therefore, it is enough to consider the product of simple polynomials output by EncCt with the non-simple polynomials output by EncKey, and vice versa. So we can define a new pair algorithm that outputs two matrices **E** and $\overline{\mathbf{E}}$, with the first one taking a linear combination of the products of the former type, and the second one of the latter type.

B Proving Symbolic Property for Encoding Schemes

We now present pair encoding schemes for several attribute-based encryption (ABE) predicates and regular languages, and give short and easy to verify proofs of their co-selective and selective symbolic property. This leads to *fully* secure ABE schemes under the $\mathsf{q\text{-}ratio_{dsg}}$ assumption, by first augmenting the encoding (if need be) so that it satisfies enhanced symbolic property (Theorem 5.2) and then applying the generic transformation from Section 5.4. Thus our approach demonstrates how the process of designing fully secure encryption schemes for even sophisticated predicates can be greatly simplified.

Specifically, we consider five types of predicates: CP-ABE with unbounded attribute re-use, CP-ABE with short ciphertexts, unbounded KP-ABE for large universes, KP-ABE with short ciphertexts, and the regular language predicate. The first pair encoding is new, the second one is from Agrawal and Chase [AC16], and the rest have been adapted from Attrapadung [Att14a]. We do not provide a pairing algorithm or prove correctness for any of the schemes, since it is either straightforward or follows easily from the cited papers.

A careful reader would observe that we do not have as many polynomials in the encodings adapted from Attrapadung as he originally had. The primary reason for this is that we have a more structured definition of pair encodings where non-lone variables are not treated as polynomials. Much like Attrapadung, both the number of non-lone variables and polynomials contribute to the size of ciphertexts and keys in the transformation of pair encodings to encryption schemes (see Remark 5.9).

Attribute-based encryption. In an attribute-based encryption scheme, the access policy is represented by a linear secret sharing (LSS) scheme $(\mathbf{A}, \pi)$, where $\mathbf{A}$ is an $n_1 \times n_2$ matrix with entries in $\mathbb{Z}_N$ and π is a mapping from $[n_1]$ to an attribute universe $\mathcal{U}$. Let $\mathbf{a_i}$ denote the ith row of $\mathbf{A}$. Let $S \subseteq \mathcal{U}$ be a set of attributes and $\Lambda = \{i \mid i \in [n_1], \pi(i) \in S\}$ be the indices of rows in $\mathbf{A}$ associated with S. Also, $a_{i,j}$ denotes the element in the ith row and jth column of $\mathbf{A}$.

We say that the LSS scheme $(\mathbf{A}, \pi)$ accepts S if $\mathbf{e} = (1, 0, \ldots, 0)$ lies in the span of rows associated with S. In other words, there exists constants $\{\varepsilon_i\}_{i\in\Lambda}$ such that $\sum_{i\in\Lambda} \varepsilon_i \mathbf{a_i} = \mathbf{e}$ if S is acceptable. On the other hand, if $(\mathbf{A}, \pi)$ does not accept (or rejects) S, then there must exist a vector $\mathbf{w} = (w_1, \ldots, w_{n_2})$ such that $\mathbf{w}$ is orthogonal to $\mathbf{a}_i$ for all $i \in S$ but not to $\mathbf{e}$. Thus we can assume that $w_1 = 1$ without loss of generality.

Throughout this section we will use some special matrices and vectors that have only one non-zero entry. For $d_1, d_2 \in \mathbb{N}$, $i \in [d_1]$ and $j \in [d_2]$, let

- $\mathbf{E}_{i,j}$ be a $d_1 \times d_2$ matrix with 1 at ith row and jth column, and 0 everywhere else;
- $\mathbf{e}_j$ be a vector of size d_2 with 1 at the jth position and 0 everywhere else;
- $\overline{\mathbf{e}}_i$ be a vector of size d_1 with 1 at the ith position and 0 everywhere else.

Note that $\mathbf{E}_{i,j}\mathbf{e}_k^\top = \overline{\mathbf{e}}_i^\top$ if $j = k$, otherwise the product is $\mathbf{0}^\top$. Similarly, $\overline{\mathbf{e}}_k\mathbf{E}_{i,j} = \mathbf{e}_j$ if $k = i$, otherwise the product is $\mathbf{0}$.

B.1 CP-ABE with Unbounded Attribute Re-Use

The predicate family in this case is indexed by $\kappa = (N, T)$. $\mathcal{X}_\kappa$ is the set of all LSS schemes where the matrix has entries in $\mathbb{Z}_N$ and the range of the mapping is $[T]$, and $\mathcal{Y}_\kappa$ is given by the set $\{S \mid S \subseteq [T]\}$. For all $(\mathbf{A}, \pi) \in \mathcal{X}_\kappa$ and $S \in \mathcal{Y}_\kappa$, $P_\kappa((\mathbf{A}, \pi), S) = 1$ if and only if $(\mathbf{A}, \pi)$ accepts S. It is clear from the definition of the predicate family that the attribute universe is $[T]$ and π need not be injective.

We need to set-up some more notation for this encoding. For any LSS scheme $(\mathbf{A}, \pi)$, let $\rho(i) = |\{j \mid \pi(j) = \pi(i), j \leq i\}|$. (If an attribute y is attached to the second and fifth rows, then $\rho(2) = 1$ and $\rho(5) = 2$.) Also, let $\sigma(y, \ell)$ denote the index of the the row which has the ℓth occurrence of y. We are now ready to formally describe a new encoding $\Pi_{\text{re-use}}$:

- $\mathsf{Param}(\mathsf{par}) \to T + 1$. Let $\mathbf{b} = (b_1, b_2, \ldots, b_T, b')$.

- $\mathsf{EncCt}((\mathbf{A},\pi),N) \to \mathbf{c}(\mathbf{s},\hat{\mathbf{s}},\mathbf{b}) = (c_1,\ldots,c_{n_1})$ where

$$c_i = \mathbf{a}_i (s_0 b', \hat{s}_2, \ldots, \hat{s}_{n_2})^\top + s_{\rho(i)} b_{\pi(i)},$$

  $\mathbf{s} = (s_0, s_1, \ldots, s_d)$, $\hat{\mathbf{s}} = (\hat{s}_2, \ldots, \hat{s}_{n_2})$, and d is the maximum number of times any attribute appears in π.

- $\mathsf{EncKey}(S,N) \to \mathbf{k}(\mathbf{r},\hat{\mathbf{r}},\mathbf{b}) = (k_1, \{k_{2,y}\}_{y\in S})$ where

$$k_1 = \alpha + r b', \qquad k_{2,y} = r b_y,$$

  $\mathbf{r} = (r)$, and $\hat{\mathbf{r}} = (\alpha)$.

In this encoding scheme, $\hat{s}_2, \ldots, \hat{s}_{n_2}, \alpha$ are lone variables while $s_0, s_1, \ldots, s_d, r$ are non-lone. We now prove selective and co-selective symbolic property. We use the notation $p : q$ to denote a variable p and its corresponding matrix/vector q.

Selective symbolic property. We use $\mathbf{E}_{i,j}$, $\mathbf{e}_j$ and $\overline{\mathbf{e}}_i$ with d_1 set to n_2 and d_2 set to d. The matrices/vectors generated by EncB, EncS and EncR are given by

$$b_y : -\sum_{\ell\in[d]} \sum_{j\in[n_2]} a_{\sigma(y,\ell),j} \mathbf{E}_{j,\ell} \text{ for } y \in [T], \qquad b' : \mathbf{E}_{1,1},$$

$$s_0 : \mathbf{e}_1, \qquad \hat{s}_j : \overline{\mathbf{e}}_j \text{ for } j = 2, \ldots, n_2, \qquad s_\ell : \mathbf{e}_\ell \text{ for } \ell \in [d],$$

$$\alpha : \mathbf{e}_1, \qquad r : -\sum_{j\in[n_2]} w_j \overline{\mathbf{e}}_j.$$

If there is no i such that $\pi(i) = y$, then $b_y = \mathbf{0}$. Note that since we are proving selective symbolic property, only the vectors output by EncR can depend on both $(\mathbf{A},\pi)$ and S.

Upon substitution in $k_{2,y}$ for $y \in S$ (and $b_y \neq \mathbf{0}$), we have

$$\left(\sum_j w_j \overline{\mathbf{e}}_j\right)\left(\sum_{\ell\in[d]} \sum_j a_{\sigma(y,\ell),j} \mathbf{E}_{j,\ell}\right) \quad = \quad \sum_{\ell\in[d]} a_{\sigma(y,\ell),j} w_j \mathbf{e}_\ell \quad = \quad \mathbf{0}.$$

It is easy to see that k_1 goes to $\mathbf{0}$ as well because $w_1 = 1$. Lastly, substitution in c_i gives

$$\sum_j a_{i,j} \overline{\mathbf{e}}_j^\top - \left(\sum_\ell \sum_j a_{\sigma(\pi(i),\ell),j} \mathbf{E}_{j,\ell}\right) \mathbf{e}_{\rho(i)}^\top \quad = \quad \sum_j a_{i,j} \overline{\mathbf{e}}_j^\top - \sum_j a_{i,j} \overline{\mathbf{e}}_j^\top \quad = \quad \mathbf{0}^\top,$$

because the ith row has $\rho(i)$th occurrence of $\pi(i)$. Further, the vectors corresponding to both α and s_0 are $\mathbf{e}_1$. Hence their inner product is non-zero.

Co-selective symbolic property. Here d_1 and d_2 are set to 1 and T, respectively. We have

$$b_y : \mathbf{0} \text{ for } y \in S \text{ and } -\mathbf{E}_{1,y} \text{ otherwise}, \qquad b' : \mathbf{E}_{1,1},$$

$$s_0 : w_1 \mathbf{e}_1, \qquad \hat{s}_j : w_j \overline{\mathbf{e}}_1 \text{ for } j = 2, \ldots, n_2, \qquad s_\ell : \sum_{i:\rho(i)=\ell} \mathbf{a}_i \mathbf{w}^\top \mathbf{e}_{\pi(i)} \text{ for } \ell \in [d],$$

$$\alpha : \mathbf{e}_1, \qquad r : -\overline{\mathbf{e}}_1.$$

Upon substitution in k_1, $k_{2,y}$ for $y \in S$, and c_i for $i \in \Lambda$, we clearly get $\mathbf{0}$. The only remaining case is c_i for $i \notin \Lambda$, for which we have

$$\mathbf{a}_i \mathbf{w}^\top \overline{\mathbf{e}}_1^\top - \mathbf{E}_{1,\pi(i)} \left(\sum_{t:\rho(t)=\rho(i)} \mathbf{a}_t \mathbf{w}^\top \mathbf{e}_{\pi(t)}^\top\right) \quad = \quad \mathbf{a}_i \mathbf{w}^\top \overline{\mathbf{e}}_1^\top - \mathbf{a}_i \mathbf{w}^\top \overline{\mathbf{e}}_1^\top \quad = \quad \mathbf{0}^\top.$$

Further, the vector corresponding to α is $\mathbf{e}_1$, and the vector for s_0 is also $\mathbf{e}_1$ since $w_1 = 1$. Hence their inner product is non-zero.

B.2 Unbounded KP-ABE with Large Universes

The predicate family in this case is indexed by $\kappa = (N)$. $\mathcal{X}_\kappa$ is given by the set $\{S \mid S \subseteq \mathbb{Z}_N\}$ and $\mathcal{Y}_\kappa$ is the set of all LSS schemes where the matrix has entries in $\mathbb{Z}_N$ and the range of the mapping is also $\mathbb{Z}_N$. For all $S \in \mathcal{X}_\kappa$ and $(\mathbf{A}, \pi) \in \mathcal{Y}_\kappa$, $P_\kappa(S, (\mathbf{A}, \pi)) = 1$ if and only if $(\mathbf{A}, \pi)$ accepts S. It is clear from the definition of the predicate family that the attribute universe is unbounded and π need not be injective.

- $\mathsf{Param}(\mathsf{par}) \to 3$. Let $\mathbf{b} = (b_0, b_1, b_2)$.
- $\mathsf{EncCt}(S, N) \to \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}) = (\{c_y\}_{y \in S})$ where

$$c_y = sb_2 + s_y(b_0 + b_1 y)$$

  and $\mathbf{s} = (s, \{s_y\}_{y \in S})$.
- $\mathsf{EncKey}((\mathbf{A}, \pi), N) \to \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}) = (\{k_{1,i}, k_{2,i}\}_{i \in [n_1]})$ where

$$k_{1,i} = \mathbf{a}_i(\alpha, v_2, \ldots, v_{n_2})^\top + r_i b_2, \qquad k_{2,i} = r_i(b_0 + b_1\pi(i)),$$

  $\mathbf{r} = (r_1, \ldots, r_{n_1})$, and $\hat{\mathbf{r}} = (\alpha, v_2, \ldots, v_{n_2})$.

In this encoding scheme, all ct-enc variables s, $\{s_y\}_{y \in S}$ are non-lone. Among key-enc variables, $r_1, \ldots, r_{n_1}$ are non-lone while $\alpha, v_2, \ldots, v_{n_2}$ are lone. Also, s plays the role of s_0.

We now prove selective and co-selective symbolic property. Below, v_1 is used to denote α.

Selective symbolic property. Let $d = |S|$ and $S = (y_1, \ldots, y_d)$. d_1 and d_2 are set to $d+1$ and d, respectively. The matrices/vectors generated by EncB, EncS and EncR are given by

$$b_0 : \sum_{j=1}^{d} -\mathbf{E}_{d+1,j} + y_j \mathbf{E}_{j,j}, \qquad b_1 : -\sum_{j=1}^{d} \mathbf{E}_{j,j}, \qquad b_2 : \mathbf{E}_{d+1,1},$$

$$s : \mathbf{e}_1, \qquad s_{y_i} : \mathbf{e}_i \text{ for } i \in [d],$$

$$v_i : w_i \mathbf{e}_1 \text{ for } i \in [n_2], \qquad r_i : \mathbf{a}_i \mathbf{w}^\top \left(-\bar{\mathbf{e}}_{d+1} + \sum_{j=1}^{d} \frac{-\bar{\mathbf{e}}_j}{y_j - \pi(i)} \right) \text{ if } i \notin \Lambda,$$

and $\mathbf{0}$ otherwise.

Upon substitution in c_{y_i} we have

$$\begin{aligned} & \mathbf{E}_{d+1,1}\mathbf{e}_1^\top + \left(\sum_{j=1}^{d} (y_j - y_i)\mathbf{E}_{j,j} - \mathbf{E}_{d+1,j} \right) \mathbf{e}_i^\top \\ = \quad & \bar{\mathbf{e}}_{d+1}^\top + (y_i - y_i)\bar{\mathbf{e}}_i^\top - \bar{\mathbf{e}}_{d+1}^\top \quad = \quad \mathbf{0}^\top. \end{aligned}$$

Substitution in $k_{1,i}$ for $i \notin \Lambda$ gives

$$\begin{aligned} & \mathbf{a}_i \mathbf{w}^\top \mathbf{e}_1 + \mathbf{a}_i \mathbf{w}^\top \left(-\bar{\mathbf{e}}_{d+1} + \sum_{j=1}^{d} \frac{-\bar{\mathbf{e}}_j}{y_j - \pi(i)} \right) \mathbf{E}_{d+1,1} \\ = \quad & \mathbf{a}_i \mathbf{w}^\top \mathbf{e}_1 - \mathbf{a}_i \mathbf{w}^\top \mathbf{e}_1 \quad = \quad \mathbf{0}. \end{aligned}$$

Finally, substitution in $k_{2,i}$ for $i \notin \Lambda$ yields

$$\begin{aligned} & \mathbf{a}_i \mathbf{w}^\top \left(-\overline{\mathbf{e}}_{d+1} + \sum_{j=1}^{d} \frac{-\overline{\mathbf{e}}_j}{y_j - \pi(i)} \right) \left(\sum_{j=1}^{d} (y_j - \pi(i)) \mathbf{E}_{j,j} - \mathbf{E}_{d+1,j} \right) \\ = \quad & \mathbf{a}_i \mathbf{w}^\top \left(\sum_{j=1}^{d} \mathbf{e}_j - \sum_{j=1}^{d} \mathbf{e}_j \frac{y_j - \pi(i)}{y_j - \pi(i)} \right) \quad = \quad \mathbf{0}. \end{aligned}$$

One could also verify with ease that $k_{1,i}$ and $k_{2,i}$ give $\mathbf{0}$ for $i \in \Lambda$ because $\mathbf{a}_i \mathbf{w}^\top = 0$ for those i. Further, the vector corresponding to α is $\mathbf{e}_1$ since $w_1 = 1$, and the vector for s_0 (or s) is also $\mathbf{e}_1$. Hence the desired inner product is non-zero.

Co-selective symbolic property. Here d_1 and d_2 are set to n_1 and $\max(n_1, n_2)$, respectively. We have

$$b_0 : -\sum_{\ell=1}^{n_1} \mathbf{E}_{\ell,\ell} \pi(\ell), \qquad b_1 : \sum_{\ell=1}^{n_1} \mathbf{E}_{\ell,\ell}, \qquad b_2 : -\sum_{\ell=1}^{n_1} \sum_{j=1}^{n_2} a_{\ell,j} \mathbf{E}_{\ell,j},$$

$$s : \sum_{j=1}^{n_2} w_j \mathbf{e}_j, \qquad s_{y_i} : \sum_{t \notin \Lambda} \frac{\mathbf{a}_t \mathbf{w}^\top \mathbf{e}_t}{y_i - \pi(t)} \text{ for } i \in [d],$$

$$r_i : \overline{\mathbf{e}}_i \text{ for } i \in [n_1], \qquad v_i : \mathbf{e}_i \text{ for } i \in [n_2].$$

Upon substitution in c_{y_i} we have

$$\begin{aligned} & \left(-\sum_{\ell=1}^{n_1} \sum_{j=1}^{n_2} a_{\ell,j} \mathbf{E}_{\ell,j} \right) \sum_{j=1}^{n_2} w_j \mathbf{e}_j^\top + \left(\sum_{\ell=1}^{n_1} \mathbf{E}_{\ell,\ell} (y_i - \pi(\ell)) \right) \sum_{t \notin \Lambda} \frac{\mathbf{a}_t \mathbf{w}^\top \mathbf{e}_t^\top}{y_i - \pi(t)} \\ = \quad & -\sum_j w_j \sum_\ell a_{\ell,j} \overline{\mathbf{e}}_\ell^\top + \sum_{\ell \notin \Lambda} \mathbf{a}_\ell \mathbf{w}^\top \overline{\mathbf{e}}_\ell^\top \quad = \quad \mathbf{0}^\top. \end{aligned}$$

Substitution in $k_{1,i}$ gives $\sum_j a_{i,j} \mathbf{e}_j - \overline{\mathbf{e}}_i \sum_{\ell,j} a_{\ell,j} \mathbf{E}_{\ell,j} = \mathbf{0}$. And finally, substitution in $k_{2,i}$ yields $\overline{\mathbf{e}}_i \sum_\ell \mathbf{E}_{\ell,\ell} (\pi(i) - \pi(\ell)) = \mathbf{0}$.

B.3 KP-ABE with Short Ciphertexts

The only difference between this predicate family and the previous one is that there is a bound on the size of attribute sets. Specifically, the family is indexed by $\kappa = (N, T)$, where T is the bound.

- $\mathsf{Param}(\mathsf{par}) \to T + 3$. Let $\mathbf{b} = (b_0, b_1, \ldots, b_{T+1}, b')$.
- $\mathsf{EncCt}(S, N) \to \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}) = (c)$ where

  $$c = sb' + \tilde{s}(b_0 + b_1 z_0 + \ldots + b_{T+1} z_T),$$

  $\mathbf{s} = (s, \tilde{s})$, and z_i is the coefficient of x^i in $p(x) := \Pi_{y \in S}(x - y)$.
- $\mathsf{EncKey}((\mathbf{A}, \pi), N) \to \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}) = (\{k_{1,i}, \mathbf{k}_{2,i}\}_{i \in [n_1]})$ where

  $$k_{1,i} = \mathbf{a}_i (\alpha, v_2, \ldots, v_{n_2})^\top + r_i b', \quad \mathbf{k}_{2,i} = \big(r_i b_0, r_i (b_2 - b_1 \pi(i)), \ldots, r_i (b_{T+1} - b_1 \pi(i)^T)\big),$$

  $\mathbf{r} = (r_1, r_2, \ldots, r_{n_1})$, and $\hat{\mathbf{r}} = (\alpha, v_2, \ldots, v_{n_2})$.

Selective. Once again, let $d = |S|$ and $S = (y_1, \ldots, y_d)$. Here, $\mathbf{E}_{i,j}, \mathbf{e}_j, \overline{\mathbf{e}}_i$ are defined in a slightly different way. For $i \in [T+1]^+$, let $\mathbf{E}_{i,1}$ be a $(T+2) \times 1$ matrix (or a column vector) with 1 at $(i+1)$th row and first column (0 everywhere else), and let $\overline{\mathbf{e}}_i$ be a $(T+2)$-length unit vector with 1 at the $(i+1)$th position. Also, let $\mathbf{e}_1 = (1)$. We substitute the variables with the following.

$$b_0 : \mathbf{E}_{T+1,1} - \sum_{t'=0}^{T} \mathbf{E}_{t',1} z_{t'}, \qquad b_{t+1} : \mathbf{E}_{t,1} \text{ for } t \in [T]^+, \qquad b' : -\mathbf{E}_{T+1,1},$$

$$s : \mathbf{e}_1, \qquad \tilde{s} : \mathbf{e}_1,$$

$$v_i : w_i \mathbf{e}_1 \text{ for } i \in [n_2], \qquad r_i : \mathbf{a}_i \mathbf{w}^\top \left(\overline{\mathbf{e}}_{T+1} + \sum_{t=0}^{T} \frac{\pi(i)^t \overline{\mathbf{e}}_t}{p(\pi(i))} \right) \text{ if } i \notin \Lambda,$$

and $\mathbf{0}$ otherwise.

Upon substitution in c we have

$$-\mathbf{E}_{T+1,1} \mathbf{e}_1^\top + \left(\mathbf{E}_{T+1,1} - \sum_{t'=0}^{T} \mathbf{E}_{t',1} z_{t'} + \sum_{t=0}^{T} \mathbf{E}_{t,1} z_t \right) \mathbf{e}_1^\top \quad = \quad \mathbf{0}^\top.$$

Rest of the analysis is for $i \notin \Lambda$; the other case is easy to see. Substitution in $k_{1,i}$ gives $\mathbf{a}_i \mathbf{w}^\top \mathbf{e}_1 - \mathbf{a}_i \mathbf{w}^\top \mathbf{e}_1 \quad = \quad 0$. The first component of $\mathbf{k}_{2,i}$ gives

$$\mathbf{a}_i \mathbf{w}^\top \left(\mathbf{e}_1 - \sum_{t'=0}^{T} \frac{\pi(i)^{t'} z_{t'} \mathbf{e}_1}{p(\pi(i))} \right) \quad = \quad \mathbf{a}_i \mathbf{w}^\top \left(\mathbf{e}_1 - \mathbf{e}_1 \frac{p(\pi(i))}{p(\pi(i))} \right) \quad = \quad \mathbf{0}.$$

And finally, substitution in the $t+1$th component of $\mathbf{k}_{2,i}$ for $t \in [T]$ yields

$$\mathbf{a}_i \mathbf{w}^\top \left(\frac{\pi(i)^t \mathbf{e}_1}{p(\pi(i))} - \frac{\mathbf{e}_1}{p(\pi(i))} \pi(i)^t \right),$$

which is also $\mathbf{0}$.

Co-selective. Here d_1 and d_2 are set to n_1 and $\mathsf{max}(n_1, n_2)$, respectively.[17] Matrices and vectors corresponding to the variables are as follows:

$$b_0 : \mathbf{0}, \qquad b_t : \sum_{\ell=1}^{n_1} \mathbf{E}_{\ell,\ell} \pi(\ell)^{t-1} \text{ for } t \in [T+1], \qquad b' : -\sum_{\ell=1}^{n_1} \sum_{j=1}^{n_2} a_{\ell,j} \mathbf{E}_{\ell,j},$$

$$s : \sum_{j=1}^{n_2} w_j \mathbf{e}_j, \qquad \tilde{s} : \sum_{\ell \notin \Lambda} \frac{\mathbf{a}_\ell \mathbf{w}^\top \mathbf{e}_\ell}{p(\pi(\ell))},$$

$$r_i : \overline{\mathbf{e}}_i \text{ for } i \in [n_1], \qquad v_i : \mathbf{e}_i \text{ for } i \in [n_2].$$

Upon substitution in c we have

$$\begin{aligned}
&\left(-\sum_{\ell=1}^{n_1} \sum_{j=1}^{n_2} a_{\ell,j} \mathbf{E}_{\ell,j} \right) \sum_{j=1}^{n_2} w_j \mathbf{e}_j^\top \quad + \quad \left(\sum_{t=1}^{T+1} z_{t-1} \sum_{\ell=1}^{n_1} \mathbf{E}_{\ell,\ell} \pi(\ell)^{t-1} \right) \sum_{\ell \notin \Lambda} \frac{\mathbf{a}_\ell \mathbf{w}^\top \mathbf{e}_\ell^\top}{p(\pi(\ell))} \\
&= -\sum_j w_j \sum_\ell a_{\ell,j} \overline{\mathbf{e}}_\ell^\top \quad + \quad \left(\sum_{\ell=1}^{n_1} \mathbf{E}_{\ell,\ell} p(\pi(\ell)) \right) \sum_{\ell \notin \Lambda} \frac{\mathbf{a}_\ell \mathbf{w}^\top \mathbf{e}_\ell^\top}{p(\pi(\ell))} \\
&= -\sum_\ell \mathbf{a}_\ell \mathbf{w}^\top \overline{\mathbf{e}}_\ell^\top \quad + \quad \sum_{\ell \notin \Lambda} \mathbf{a}_\ell \mathbf{w}^\top \overline{\mathbf{e}}_\ell^\top \quad = \quad \mathbf{0}^\top.
\end{aligned}$$

Substitution in $k_{1,i}$ gives $\sum_j a_{i,j} \mathbf{e}_j - \overline{\mathbf{e}}_i \sum_{\ell,j} a_{\ell,j} \mathbf{E}_{\ell,j} = \mathbf{0}$. And finally, substitution in the tth component of $\mathbf{k}_{2,i}$ for $t \in [2, T+1]$ yields $\overline{\mathbf{e}}_i \sum_\ell \mathbf{E}_{\ell,\ell} (\pi(\ell)^{t-1} - \pi(i)^{t-1}) = \mathbf{0}$.

[17] $\mathbf{E}_{i,j}, \mathbf{e}_i, \overline{\mathbf{e}}_j$ are *not* defined as in the selective property. Rather we use the same definition that we have been using for most of the proofs so far (see the paragraph on attribute-based encryption, just before Appendix B.1.)

B.4 CP-ABE with Short Ciphertexts

The predicate family in this case is indexed by $\kappa = (N, T, n_1, n_2)$. $\mathcal{X}_\kappa$ is given by the set $\{S \mid S \subseteq [T]\}$. $\mathcal{Y}_\kappa$ is the set of all LSS schemes where the matrix is of size $n_1 \times n_2$ with entries in $\mathbb{Z}_N$ and the range of the mapping is also $\mathbb{Z}_N$. (So there is a bound on the size of attribute sets but not on the universe.) For all $S \in \mathcal{X}_\kappa$ and $(\mathbf{A}, \pi) \in \mathcal{Y}_\kappa$, $P_\kappa(S, (\mathbf{A}, \pi)) = 1$ iff $(\mathbf{A}, \pi)$ accepts S.

- $\mathsf{Param}(\mathsf{par}) \to n_1(n_2 + T + 1)$. Let $\mathbf{b} = \left(\left\{b_{i,j}\right\}_{i\in[n_1],j\in[n_2]}, \left\{b'_{i,t}\right\}_{i\in[n_1],t\in[T]^+}\right)$.
- $\mathsf{EncCt}((A, \pi), N) \to \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}) = (c)$ where

$$c \;\; = \;\; s\left(\sum_{\substack{i\in[n_1],\\ j\in[n_2]}} a_{i,j} b_{i,j} + \sum_{\substack{i\in[n_1],\\ t\in[T]^+}} \pi(i)^t b'_{i,t}\right),$$

  and $\mathbf{s} = (s)$.
- $\mathsf{EncKey}(S, N) \to \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}) = \left(\left\{k_{2,i,j}, k_{3,i,\ell,j}, k_{4,i,y}, k_{5,i,\ell,t}\right\}_{i,\ell\in[n_1],i\neq\ell,j\in[n_2],y\in S,t\in[T]^+}\right)$ where

$$k_{1,i,j} = r_i b_{i,j} - v_j, \qquad k_{2,i,\ell,j} = r_i b_{\ell,j},$$

$$k_{3,i,y} = r_i \sum_{t\in[0,T]} y^t b'_{i,t}, \qquad k_{4,i,\ell,t} = r_i b'_{\ell,t},$$

  $\mathbf{r} = (r_1, r_2, \ldots, r_{n_1})$, and $\hat{\mathbf{r}} = (v_1, v_2, \ldots, v_{n_2})$.

Selective. Let $p(x) = \Pi_{y\in S}(x - y)$ be a polynomial of degree at most T with attributes as the roots, and z_i be the coefficient of x^i in it for $i \in [T]^+$. Here d_1 and d_2 are set to $n_1(n_2 + T + 1)$ and 1, respectively. We have

$$b_{i,j} : \mathbf{E}_{(i-1)n_2+j,1} \text{ for } i \in [n_1], j \in [n_2], \qquad b'_{i,t} : \mathbf{E}_{n_1 n_2+(i-1)(T+1)+t+1,1} \text{ for } i \in [n_1], t \in [T],$$

$$b'_{i,0} : -\sum_{t\in[T]} \pi(i)^t \mathbf{E}_{n_1 n_2+(i-1)(T+1)+t+1,1} - \sum_{j\in[n_2]} a_{i,j} \mathbf{E}_{(i-1)n_2+j,1} \text{ for } i \in [n_1],$$

$$s : \mathbf{e}_1, \qquad v_j : w_j \mathbf{e}_1 \text{ for } j \in [n_2],$$

$$r_i : \sum_{j\in[n_2]} w_j \bar{\mathbf{e}}_{(i-1)n_2+j} - \frac{\mathbf{a}_i \mathbf{w}^\mathsf{T}}{p(\pi(i))} \sum_{t\in[T]^+} z_t \cdot \bar{\mathbf{e}}_{n_1 n_2+(i-1)(T+1)+t+1} \text{ for } i \in [n_1],$$

for $i \notin \Lambda$. (When $i \in \Lambda$, we just have the first term.)

Substituting these values in c, we have

$$\left(\sum_{i,j} a_{i,j} \mathbf{E}_{(i-1)n_2+j,1} + \sum_{i,t} \pi(i)^t \mathbf{E}_{n_1 n_2+(i-1)(T+1)+t+1,1}\right.$$
$$\left. - \sum_{i,t} \pi(i)^t \mathbf{E}_{n_1 n_2+(i-1)(T+1)+t+1,1} - \sum_{i,j} a_{i,j} \mathbf{E}_{(i-1)n_2+j,1}\right) \mathbf{e}_1^\mathsf{T},$$

which is clearly $\mathbf{0}$. Verifying $k_{1,i,j}$, $k_{2,i,\ell,j}$, and $k_{4,i,\ell,t}$ is straightforward. Finally, $k_{3,i,y}$ upon substitution (for $i \notin \Lambda$) gives

$$\left(\sum_j w_j \bar{\mathbf{e}}_{(i-1)n_2+j} - \frac{\mathbf{a}_i\mathbf{w}^\top}{p(\pi(i))}\sum_t z_t \cdot \bar{\mathbf{e}}_{n_1n_2+(i-1)(T+1)+t+1}\right)\cdot$$

$$\left(\sum_t \left(y^t - \pi(i)^t\right)\mathbf{E}_{n_1n_2+(i-1)(T+1)+t+1,1} - \sum_j a_{i,j}\mathbf{E}_{(i-1)n_2+j,1}\right)$$

$$= -\sum_j w_j a_{i,j}\mathbf{e}_1 - \frac{\mathbf{a}_i\mathbf{w}^\top}{p(\pi(i))}\mathbf{e}_1\sum_t z_t\left(y^t - \pi(i)^t\right) \quad = \quad -\mathbf{a}_i\mathbf{w}^\top\mathbf{e}_1 + \frac{\mathbf{a}_i\mathbf{w}^\top}{p(\pi(i))}\mathbf{e}_1 p(\pi(i)) \quad = \quad \mathbf{0}.$$

Co-selective. Let p be the same polynomial as defined for the selective property. d_1 and d_2 are set to n_1 and n_2+1, respectively. Then, we have

$$b_{i,j} : \mathbf{E}_{i,j} \text{ for } i \in [n_1], j \in [n_2], \qquad b'_{i,t} : z_t\mathbf{E}_{i,n_2+1} \text{ for } i \in [n_1], t \in [T]^+,$$

$$s : \sum_{j=1}^{n_2} w_j\mathbf{e}_j - \sum_{i\notin\Lambda}\frac{\mathbf{a}_i\mathbf{w}^\top\mathbf{e}_{n_2+1}}{p(\pi(i))},$$

$$r_i : \bar{\mathbf{e}}_i \text{ for } i \in [n_1], \qquad v_j : \mathbf{e}_j \text{ for } j \in [n_2].$$

Substituting these values in c, we have

$$\left(\sum_{i,j} a_{i,j}\mathbf{E}_{i,j} + \sum_{i,t}\pi(i)^t z_t\mathbf{E}_{i,n_2+1}\right)\left(\sum_j w_j\mathbf{e}_j^\top - \sum_{i\notin\Lambda}\frac{\mathbf{a}_i\mathbf{w}^\top\mathbf{e}_{n_2+1}^\top}{p(\pi(i))}\right)$$
$$= \sum_{i,j} a_{i,j}w_j\bar{\mathbf{e}}_i^\top + \left(\sum_i p(\pi(i))\mathbf{E}_{i,n_2+1}\right)\left(-\sum_{i\notin\Lambda}\frac{\mathbf{a}_i\mathbf{w}^\top\mathbf{e}_{n_2+1}^\top}{p(\pi(i))}\right)$$
$$= \sum_i \mathbf{a}_i\mathbf{w}^\top\bar{\mathbf{e}}_i^\top - \sum_{i\notin\Lambda}\mathbf{a}_i\mathbf{w}^\top\bar{\mathbf{e}}_i^\top \quad = \quad \mathbf{0}^\top.$$

Substitution in $k_{1,i,j}$ gives $\bar{\mathbf{e}}_i\mathbf{E}_{i,j} - \mathbf{e}_j = \mathbf{0}$. Verifying $k_{2,i,\ell,j}$ and $k_{4,i,\ell,t}$ is straightforward. Finally, $k_{3,i,y}$ upon substitution gives $\bar{\mathbf{e}}_i\sum_t y^t z_t\mathbf{E}_{i,n_2+1} = \mathbf{e}_{n_2+1}\sum_t y^t z_t = \mathbf{0}$.

B.5 Regular Languages

A deterministic finite automaton (DFA) is a 5-tuple $(Q, \Lambda, \Psi, q_0, F)$ where Q is the set of states, Λ is the alphabet set, $\Psi \subseteq Q \times Q \times \Lambda$ is the transition table, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of final states. One can, without loss of generality, assume that there is only one final state, and it has no outgoing transition. The predicate family for regular languages associates DFAs with keys and strings over Λ^* with ciphertexts. The original encryption scheme proposed by Waters [Wat12] could only operate over small universes with selective security. Attrapadung later gave a fully secure scheme over large universes under specialized assumptions—the ℓ-Expanded Diffie-Hellman Exponent Assumption-1 (EDHE1) and the (n,m)-Expanded Diffie-Hellman Exponent Assumption-2 (EDHE2). We show that a (simplified version) of his pair encoding scheme can be shown to have both selective and co-selective symbolic property, thus giving a fully secure scheme (with unbounded universe) under the q-ratio assumption (Corollary 7.2). Recall that this assumption is implied by the (n,m)-EDHE2 assumption (Theorem C.2).

Formally, the predicate family is indexed by $\kappa = (N)$. The set $\mathcal{Y}_\kappa$ has all DFAs $M = (Q = \{q_0, \ldots, q_{n-1}\}, \Lambda, \Psi, q_0, q_{n-1})$ where $\Lambda = \mathbb{Z}_N$, and $\mathcal{X}_\kappa = \{\mathbf{w} \mid \mathbf{w} \in (\mathbb{Z}_N)^*\}$. A string $\mathbf{w} = (w_1, \ldots, w_\ell)$

is accepted by M if there exists a sequence of states $q_1, \dots, q_\ell$ s.t. $(q_{i-1}, q_i, w_i) \in \Psi$ and $q_\ell = q_{n-1}$. We now give a pair encoding scheme for this predicate family.

- $\mathsf{Param}(\mathsf{par}) \to 5$. Let $\mathbf{b} = (b_0, b_1, b_2, b_3, b_4)$.
- $\mathsf{EncCt}(\mathbf{w} = (w_1, \dots, w_\ell), N) \to \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}) = (c_1, \{c_{2,i}\}_{i \in [\ell]})$ where
$$c_1 = s_0 b_0, \qquad c_{2,i} = s_{i-1}(b_1 + b_2 w_i) + s_i(b_3 + b_4 w_i),$$
and $\mathbf{s} = (s_\ell, s_0, \dots, s_{\ell-1})$. Here s_ℓ plays the role of the special s_0 variable.
- $\mathsf{EncKey}(M = (Q, \mathbb{Z}_N, \Psi, q_0, q_{n-1}), N) \to \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}) = (k_1, \{k_{2,t}, k_{3,t}\}_{t \in [m]})$ where
$$k_1 = -u_0 + r_0 b_0, \qquad k_{2,t} = u_{x_t} + r_t(b_1 + b_2 \sigma_t), \qquad k_{3,t} = -u_{y_t} + r_t(b_3 + b_4 \sigma_t),$$
where $\Psi = \big\{(q_{x_t}, q_{y_t}, \sigma_t) \mid t \in [m]\big\}$, $\mathbf{r} = (r_0, \dots, r_m)$, and $\hat{\mathbf{r}} = (u_{n-1}, u_0, \dots, u_{n-2})$. Here u_{n-1} plays the role of the special α variable.

The proof of selective and co-selective symbolic properties are as follows.

Selective. Here, $\mathbf{E}_{i,j}, \mathbf{e}_j, \overline{\mathbf{e}}_i$ are defined in a slightly different way, for $i \in [3\ell+1]$, $j \in [\ell]^+$. Let $\mathbf{E}_{i,j}$ be a $(3\ell+1) \times (\ell+1)$ matrix with 1 at ith row and $(j+1)$th column, and 0 everywhere else. Also, let $\mathbf{e}_j$ be a $(\ell+1)$-length unit vector with 1 at $(j+1)$th position and $\overline{\mathbf{e}}_i$ be a $(3\ell+1)$-length unit vector with 1 at the ith position. Matrices for the common variables are

$$b_0 : \sum_{j=1}^{\ell} \mathbf{E}_{j,j}, \qquad b_1 : \sum_{j=1}^{\ell} \mathbf{E}_{j,j-1}, \qquad b_2 : \mathbf{0},$$

$$b_3 : \sum_{j=1}^{\ell} -w_j \mathbf{E}_{\ell+j,j} - \mathbf{E}_{j,j} - \sum_{j=0}^{\ell} w_j \mathbf{E}_{2\ell+j+1,j}, \qquad b_4 : \sum_{j=1}^{\ell} \mathbf{E}_{\ell+j,j} + \sum_{j=0}^{\ell} \mathbf{E}_{2\ell+j+1,j}.$$

We just have $s_i : \mathbf{e}_i$ for $i \in [0, \ell]$ in the case of ct-enc variables.

Following [Wat12, Att14a], we define some notation. Let $\mathbf{w}_i$ denote the vector formed by the last $\ell - i$ symbols of $\mathbf{w}$ for $i \in [0, \ell]$, so that $\mathbf{w}_0 = \mathbf{w}$ and $\mathbf{w}_\ell$ is empty. Let M_k be the same DFA as M except that the start state is set to q_k, for $k \in [0, n-1]$. Now, define V_k to be the set of all i such that the DFA M_k accepts $\mathbf{w}_i$. One can see that $0 \notin V_0$ because $\mathbf{w}_0 = \mathbf{w}$ is not accepted by $M_0 = M$. Also, $\ell \notin V_k$ for any $k \in [0, n-2]$ and $V_{n-1} = \{\ell\}$ since q_{n-1} is the only final state and it has no outgoing transition. If we define $V_x^{+1} = \{i+1 \mid i \in V_x\}$ for $q_x \in Q$, then it can be shown that for all $(q_x, q_y, \sigma) \in \Psi$, if $i \in (V_x^{+1} \setminus V_y) \cup (V_y \setminus V_x^{+1})$ then $\sigma \neq w_i$. We are now ready to define the vectors for key-enc variables:

$$r_t : \sum_{j \in V_{x_t}^{+1}} \overline{\mathbf{e}}_j + \sum_{j \in V_{x_t}^{+1} \setminus V_{y_t}} \frac{\overline{\mathbf{e}}_{\ell+j}}{\sigma_t - w_j} - \sum_{j \in V_{y_t} \setminus V_{x_t}^{+1}} \frac{\overline{\mathbf{e}}_{2\ell+j+1}}{\sigma_t - w_j} \quad \text{for } t \in [m],$$

$$r_0 : -\sum_{j \in V_0} \overline{\mathbf{e}}_j, \qquad u_k : -\sum_{j \in V_k} \mathbf{e}_j \quad \text{for } k \in [0, n-1].$$

We now substitute the matrices and vectors defined above into the polynomials one by one. Upon substitution in c_1, we clearly get $\mathbf{0}$, and in $c_{2,i}$,

$$\left(\sum_{j=1}^{\ell} \mathbf{E}_{j,j-1}\right) \mathbf{e}_{i-1}^{\top} + \left(\sum_{j=1}^{\ell} (w_i - w_j) \mathbf{E}_{\ell+j,j} - \mathbf{E}_{j,j} + \sum_{j=0}^{\ell} (w_i - w_j) \mathbf{E}_{2\ell+j+1,j}\right) \mathbf{e}_i^{\top}$$
$$= \quad \overline{\mathbf{e}}_i^{\top} + (w_i - w_i)\overline{\mathbf{e}}_{\ell+i}^{\top} + (w_i - w_i)\overline{\mathbf{e}}_{2\ell+i+1}^{\top} - \overline{\mathbf{e}}_i^{\top} \quad = \quad \mathbf{0}^{\top}.$$

Substitution in k_1 gives

$$\sum_{j \in V_0} \mathbf{e}_j - \sum_{j \in V_0} \overline{\mathbf{e}}_j \left(\sum_{j=1}^{\ell} \mathbf{E}_{j,j} \right) \quad = \quad \sum_{j \in V_0} \mathbf{e}_j - \sum_{j \in V_0} \mathbf{e}_j \quad = \quad \mathbf{0},$$

because $0 \notin V_0$. For $k_{2,t}$ we have

$$\begin{aligned} & -\sum_{j \in V_{x_t}} \mathbf{e}_j + \left(\sum_{j \in V_{x_t}^{+1}} \overline{\mathbf{e}}_j + \sum_{j \in V_{x_t}^{+1} \setminus V_{y_t}} \frac{\overline{\mathbf{e}}_{\ell+j}}{\sigma_t - w_j} - \sum_{j \in V_{y_t} \setminus V_{x_t}^{+1}} \frac{\overline{\mathbf{e}}_{2\ell+j+1}}{\sigma_t - w_j} \right) \sum_{j=1}^{\ell} \mathbf{E}_{j,j-1} \\ = \quad & -\sum_{j \in V_{x_t}} \mathbf{e}_j + \sum_{j \in V_{x_t}^{+1}} \mathbf{e}_{j-1} \quad = \quad \mathbf{0}, \end{aligned}$$

because $x_t \in [0, n-2]$, so $\ell \notin V_{x_t}$. Finally, for $k_{3,t}$,

$$\begin{aligned} & \sum_{j \in V_{y_t}} \mathbf{e}_j + \left(\sum_{j \in V_{x_t}^{+1}} \overline{\mathbf{e}}_j + \sum_{j \in V_{x_t}^{+1} \setminus V_{y_t}} \frac{\overline{\mathbf{e}}_{\ell+j}}{\sigma_t - w_j} - \sum_{j \in V_{y_t} \setminus V_{x_t}^{+1}} \frac{\overline{\mathbf{e}}_{2\ell+j+1}}{\sigma_t - w_j} \right) \\ & \qquad \left(\sum_{j=1}^{\ell} (\sigma_t - w_j) \mathbf{E}_{\ell+j,j} - \mathbf{E}_{j,j} + \sum_{j=0}^{\ell} (\sigma_t - w_j) \mathbf{E}_{2\ell+j+1,j} \right) \\ = \quad & \sum_{j \in V_{y_t}} \mathbf{e}_j + \left(-\sum_{j \in V_{x_t}^{+1}} \mathbf{e}_j + \sum_{j \in V_{x_t}^{+1} \setminus V_{y_t}} \mathbf{e}_j - \sum_{j \in V_{y_t} \setminus V_{x_t}^{+1}} \mathbf{e}_j \right) \quad = \quad \sum_{j \in V_{y_t}} \mathbf{e}_j - \sum_{j \in V_{y_t}} \mathbf{e}_j, \end{aligned}$$

which is also equal to $\mathbf{0}$. The inner-product we are interested here is in between the vectors for s_ℓ and u_{n-1}. They are given by $\mathbf{e}_\ell$ and $-\sum_{j \in V_{n-1}} \mathbf{e}_j = -\mathbf{e}_\ell$, respectively. Thus the inner-product is not zero as desired.

Co-selective. Without loss of generality, we can assume that $m \geq n-1$ because if there is a q_x (except q_0) such that $(q_y, q_x, \sigma) \notin \Psi$ for any q_y, σ, then q_x is an unreachable state and it can be left out. We define $\mathbf{E}_{i,j}, \mathbf{e}_j, \overline{\mathbf{e}}_i$ for $i \in [m]^+$, $j \in [3m]^+$ as follows: $\mathbf{E}_{i,j}$ has a 1 in the $(i+1)$th row and $(j+1)$th column, and 0 everywhere else; $\mathbf{e}_j$ is a $(3m+1)$-length unit vector with 1 at the $(j+1)$th position; and, $\overline{\mathbf{e}}_i$ is a $(m+1)$-length unit vector with 1 at the $(i+1)$th position.

Matrices for the common variables are

$$b_0 : \mathbf{E}_{0,0}, \qquad b_1 : \sum_{k=1}^{m} \sigma_k \mathbf{E}_{k,m+k} - \mathbf{E}_{k,x_k}, \qquad b_2 : \sum_{k=1}^{m} -\mathbf{E}_{k,m+k},$$

$$b_3 : \sum_{k=1}^{m} -\sigma_k \mathbf{E}_{k,2m+k} + \mathbf{E}_{k,y_k}, \qquad b_4 : \sum_{k=1}^{m} \mathbf{E}_{k,2m+k}.$$

We just have $r_i : \overline{\mathbf{e}}_i$ for $i \in [0, m]$ and $u_k : \mathbf{e}_k$ for $k \in [0, n-1]$ in the case of key-enc variables. Once again, we define some notation before moving to the other variables. Let $U_i = \{k \in [0, n-1] \mid M_k \text{ accepts } \mathbf{w}_i\}$ for $i \in [0, \ell]$, where (recall that) M_k is the DFA M but with q_k as the start state and $\mathbf{w}_i$ is the vector with the last $\ell - i$ symbols of $\mathbf{w}$. One can see that $0 \notin U_0$ for the same reason as $0 \notin V_0$. Also, $U_\ell = \{n-1\}$ because empty string can only be accepted if the start and final states are the same. Further, it can be shown that for all $i \in [\ell]$, if $\sigma_t = w_i$ then $x_t \in U_{i-1} \wedge y_t \in U_i$ or $x_t \notin U_{i-1} \wedge y_t \notin U_i$. Now, vectors for ct-enc variables are given as

$$s_i : \sum_{k \in U_i} \mathbf{e}_k - \sum_{\substack{x_k \in U_i, \\ \sigma_k \neq w_{i+1}}} \frac{\mathbf{e}_{m+k}}{w_{i+1} - \sigma_k} - \sum_{\substack{y_k \in U_i, \\ \sigma_k \neq w_i}} \frac{\mathbf{e}_{2m+k}}{w_i - \sigma_k}$$

for $i \in [0, \ell]$.

We now substitute the matrices and vectors defined above into polynomials one by one. Upon substitution in k_1, we clearly get $\mathbf{0}$, and in $k_{2,t}$,

$$\mathbf{e}_{x_t} + \bar{\mathbf{e}}_t \left(\sum_k (\sigma_k - \sigma_t) \mathbf{E}_{k,m+k} - \mathbf{E}_{k,x_k} \right) \quad = \quad \mathbf{e}_{x_t} + (\sigma_t - \sigma_t) \mathbf{e}_{m+t} - \mathbf{e}_{x_t} \quad = \quad \mathbf{0}.$$

Substitution in $k_{3,t}$ gives

$$-\mathbf{e}_{y_t} + \bar{\mathbf{e}}_t \left(\sum_k (\sigma_t - \sigma_k) \mathbf{E}_{k,2m+k} + \mathbf{E}_{k,y_k} \right) \quad = \quad -\mathbf{e}_{y_t} + (\sigma_t - \sigma_t) \mathbf{e}_{2m+t} + \mathbf{e}_{y_t} \quad = \quad \mathbf{0}.$$

c_1 gives us $\mathbf{0}^\top$ because $0 \notin U_0$. And lastly, for $c_{2,i}$ we have

$$\begin{aligned} & \left[\sum_k (\sigma_k - w_i) \mathbf{E}_{k,m+k} - \mathbf{E}_{k,x_k} \right] \left[\sum_{k \in U_{i-1}} \mathbf{e}_k^\top - \sum_{\substack{x_k \in U_{i-1}, \\ \sigma_k \neq w_i}} \frac{\mathbf{e}_{m+k}^\top}{w_i - \sigma_k} - \sum_{\substack{y_k \in U_{i-1}, \\ \sigma_k \neq w_{i-1}}} \frac{\mathbf{e}_{2m+k}^\top}{w_{i-1} - \sigma_k} \right] \\ & \qquad + \left[\sum_k (w_i - \sigma_k) \mathbf{E}_{k,2m+k} + \mathbf{E}_{k,y_k} \right] \left[\sum_{k \in U_i} \mathbf{e}_k^\top - \sum_{\substack{x_k \in U_i, \\ \sigma_k \neq w_{i+1}}} \frac{\mathbf{e}_{m+k}^\top}{w_{i+1} - \sigma_k} - \sum_{\substack{y_k \in U_i, \\ \sigma_k \neq w_i}} \frac{\mathbf{e}_{2m+k}^\top}{w_i - \sigma_k} \right] \\ & = \quad \sum_{\substack{x_k \in U_{i-1}, \\ \sigma_k \neq w_i}} \bar{\mathbf{e}}_k^\top - \sum_{x_k \in U_{i-1}} \bar{\mathbf{e}}_k^\top - \sum_{\substack{y_k \in U_i, \\ \sigma_k \neq w_i}} \bar{\mathbf{e}}_k^\top + \sum_{y_k \in U_i} \bar{\mathbf{e}}_k^\top \quad = \quad - \sum_{\substack{x_k \in U_{i-1}, \\ \sigma_k = w_i}} \bar{\mathbf{e}}_k^\top + \sum_{\substack{y_k \in U_i, \\ \sigma_k = w_i}} \bar{\mathbf{e}}_k^\top, \end{aligned}$$

which is also equal to $\mathbf{0}^\top$ because when $\sigma_k = w_i$, $x_k \in U_{i-1}$ if and only if $y_k \in U_i$.

It is easy to see that the inner-product of vectors corresponding to s_ℓ and u_{n-1} is non-zero because $n - 1 \in U_\ell$.

C The New Assumption

C.1 Relationship Between the Two Versions

We argue that Chen and Wee's prime order DSG construction [CW14a, Section 6.2], referred to as Π_{prime} below, satisfies the $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ assumption (Definition 5.6) if the underlying group satisfies the $\mathsf{q\text{-}ratio}$ assumption (Definition 5.7).

Fix any positive integers d_1 and d_2. Suppose $\mathcal{A}$ is a PPT adversary that can break the (d_1, d_2)-$\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ assumption. We use $\mathcal{A}$ to design an adversary $\mathcal{B}$ that can break the $\mathsf{q\text{-}ratio}$ assumption with the same parameters. $\mathcal{B}$ receives $(\mathsf{par}, D_{\mathcal{G}}, D_{\mathcal{H}}, T)$ as input where $\mathsf{par} = (N, \mathcal{G}, \mathcal{H}, \mathcal{G}_T, g, h, e)$ and T is either $\hat{h}^{1/u_0}$ or a random element from $\mathcal{H}$. It runs the SampP algorithm of Π_{prime} using par as the output of the prime-order group generator. It gives $(\mathrm{PP}, \mathrm{SP}, \mathsf{td} := (\mathbf{f}, \mathbf{f}_1, \ldots, \mathbf{f}_n, \mathbf{f}^*, \mathbf{f}_1^*, \ldots, \mathbf{f}_n^*))$ to $\mathcal{A}$ where $(\mathrm{PP}, \mathrm{SP})$ is the output of SampP and td is sampled during its run. The output of algorithms $\overline{\mathsf{SampG}}_0$ and $\overline{\mathsf{SampH}}_0$ of Π_{prime} is given by $g^{\hat{s}\mathbf{f}}$ and $h^{\hat{r}\mathbf{f}^*}$, respectively, where $\hat{s}, \hat{r} \leftarrow_R \mathbb{Z}_p^*$. Thus $g^{\hat{s}}$ and $h^{\hat{r}}$ are uniformly distributed over $\mathcal{G}$ and $\mathcal{H}$, respectively, except with probability $1/p$ in each case. So to simulate the $\mathbb{G}$ and $\mathbb{H}$ terms in the $\mathsf{q\text{-}ratio}$ assumption, $\mathcal{B}$ can simply raise the terms in $D_{\mathcal{G}}$ to $\mathbf{f}$ and the terms in $D_{\mathcal{H}}$ to $\mathbf{f}^*$. Finally, $\mathcal{A}$ is given $T^{\mathbf{f}^*}$ as the challenge. If $T = \hat{h}^{1/u_0}$, then $T^{\mathbf{f}^*}$ is identically distributed to $h^{\hat{r}/u_0 \mathbf{f}^*}$, otherwise $T^{\mathbf{f}^*}$ is a fresh sample from $\overline{\mathsf{SampH}}_0$ (except with probability $1/p$).

C.2 Relationship with Other Assumptions

Here we will show that the q-ratio assumption on prime-order bilinear maps is in fact implied by several other assumptions which have been proposed recently and used to construct ABE schemes.

We first consider the ℓ-*EDHE*2 assumption, which was proposed in [Att14a] to prove co-selective security of a construction for ABE for regular languages. The original assumption required that $\mathcal{G}$ be a prime order subgroup of a composite order group. Attrapadung later proposed using the prime order analogue to achieve a construction for ABE for regular languages in prime order groups [Att15]. We will consider the prime order analogue here.

Definition C.1 ((n,m)-*EDHE*2 Assumption in group $\mathcal{G}$ [Att14a])**.** *The assumption is as follows. Let N be the order of group $\mathcal{G}$. Let D, T_0, T_1 be sampled as follows. Choose $g \leftarrow_R \mathcal{G}$ and $a, b, c, d_1, \ldots, d_m, z \leftarrow_R \mathbb{Z}_N$. Set D to be a tuple containing the following values:*

$$g, g^a, g^b, g^{a^{n-1}c/z}$$

$$g^{a^i/d_j^2}, g^{a^i b/d_j}, g^{d_j}, g^{a^i d_j/d_{j'}^2}, g^{a^i bd_j/d_{j'}}, g^{a^i/d_j^6}, g^{a^i d_j/d_{j'}^6} \quad \forall\ i \in [n], j, j' \in [m], j \neq j'$$

$$g^{a^i c}, g^{a^i bcd_j} \quad \forall\ i \in [n-1]^+$$

$$g^{a^i bcd_j^5} \quad \forall\ i \in [n]^+, j \in [m]$$

$$g^{a^i bcd_j/d_{j'}^2}, g^{a^i bcd_j^5/d_{j'}^6} \quad \forall\ i \in [2n-1], j, j' \in [m], j \neq j'$$

$$g^{a^i bc/d_j} \quad \forall\ i \in [2n-1], i \neq n, j \in [m]$$

$$g^{a^i c/d_j^2}, g^{a^i b^2 cd_j/d_{j'}}, g^{a^i bcd_j/d_{j'}^6}, g^{a^i c/d_j^6}, g^{a^i bcd_j^5/d_{j'}^2}, g^{a^i b^2 cd_j^5/d_{j'}} \quad \forall\ i \in [2n-1], j, j' \in [m]$$

Set $T_0 = g^{abz}$ and $T_1 \leftarrow_R \mathcal{G}$.

Then the assumption is that the distributions (D, T_0) and (D, T_1) are indistinguishable to any PPT *adversary.*

Here we are concerned with asymmetric bilinear groups, so we consider the natural translation of (n,m)-*EDHE*2, in which the adversary is also given $h \leftarrow \mathcal{H}$ raised to all the same exponents. (Note that in the generic group model, this is hard if the above symmetric group assumption is hard.)

Theorem C.2. *The asymmetric (n,m)-EDHE2 assumption implies the (n,m)-*q-ratio *assumption.*

Proof. Suppose we have an adversary $\mathcal{A}$ for the (n,m)-q-ratio assumption. Then we construct an adversary $\mathcal{B}$ for the (n,m)-*EDHE*2 assumption as follows. $\mathcal{B}$ receives (D, T), and ignores all values in D except the values of the form $g^{a^{n-1}c/z}, g^{d_j}, g^{a^i bd_j/d_{j'}}, h^{a^i c}, h^{a^i bc/d_j}$.

It will implicitly set the generators in the q-ratio assumption to be $(\bar{g}, \bar{h}) = (g^{a^{n-1}c}, h^{ab})$. It will also choose random $w_1, \ldots, w_n \leftarrow \mathbb{Z}_N$ and implicitly set $u_0 = 1/z$, $u_j = \frac{d_j}{ca^{n-1}}$, $v_i = \frac{w_i a^{i-2} c}{b}$ for $i \in [n], j \in [m]$. Finally, it sends $\mathcal{A}$ a challenge consisting of

$$g^{a^{n-1}c/z}, \{g^{d_j}\}_{j\in[m]}, \{(g^{a^{n-k+1}bd_i/d_j})^{1/w_k}\}_{i,j\in[m], i\neq j, k\in[n]};$$

$$\{(h^{a^{i-1}c})^{w_i}\}_{i\in[n]}, \{(h^{a^{n+i-j}bc/d_k})^{w_i/w_j}\}_{i,j\in[n], i\neq j, k\in[m]};$$

and

$$T := h^{abz}$$

It then outputs whatever $\mathcal{A}$ does.

To see that this is a perfect simulation of the adversary's view in the q-ratio game, consider the following:

$$g^{a^{n-1}c/z} = (g^{a^{n-1}c})^{1/z} = \bar{g}^{u_0}$$

$$g^{d_j} = (g^{a^{n-1}c})^{\frac{d_j}{a^{n-1}c}} = \bar{g}^{u_j}$$

$$(g^{a^{n-k+1}bd_i/d_j})^{1/w_k} = (g^{a^{n-1}c})^{\frac{a^{-(k-2)}b}{cw_k}d_i/d_j} = \bar{g}^{\frac{u_i}{u_j v_k}}$$

$$(h^{a^{i-1}c})^{w_i} = (h^{ab})^{w_i a^{i-2}c/b} = \bar{h}^{v_i}$$

$$(h^{a^{n+i-j}bc/d_k})^{w_i/w_j} = (h^{ab})^{\frac{a^{i-j}a^{n-1}cw_i}{d_k w_j}} = \bar{h}^{\frac{v_i}{v_j u_k}}$$

Finally,

$$h^{abz} = (h^{ab})^z = \bar{h}^{1/u_0},$$

so if $\mathcal{B}$ is given a challenge with h^{abz}, $\mathcal{A}$'s view will be identical to that in q-ratio when he is given T_0, and if $\mathcal{B}$'s challenge is random, $\mathcal{A}$'s view will be identical to that in q-ratio when he is given T_1. □

Next we consider the source group q-parallel BDHE assumption, which was proposed in [LW12] to prove security of a ciphertext policy ABE scheme. (This scheme is interesting because, unlike all of the CP-ABE's proved fully secure under non-q-type assumptions, it allows for policies in which the same attribute appears many times, without trivially blowing up the key sizes.)

Definition C.3 (q-parallel BDHE Assumption in group $\mathcal{G}$)**.** *The assumption is as follows. Let p be the order of group $\mathcal{G}$. Let D, T_0, T_1 be sampled as follows. Choose $g \leftarrow_R \mathcal{G}$ and $c, d, f, b_1, \ldots, b_q \leftarrow_R \mathbb{Z}_p$. Set D to be a tuple containing the following values:*

$$\begin{array}{rl} g, g^f, g^{df} & \\ g^{c^i} & \forall\ i \in [2q], i \neq q+1 \\ g^{c^i/b_j} & \forall\ i \in [2q], j \in [q], i \neq q+1 \\ g^{dfb_j} & \forall\ j \in [q] \\ g^{dfc^i b_{j'}/b_j} & \forall\ i \in [q], j, j' \in [q], j \neq j' \end{array}$$

Set $T_0 = g^{dc^{q+1}}$ and $T_1 \leftarrow_R \mathcal{G}$.

Then the assumption is that the distributions (D, T_0) and (D, T_1) are indistinguishable to any PPT *adversary.*

Here we are concerned with asymmetric bilinear groups, so we consider the natural translation of q-parallel BDHE assumption, in which the adversary is also given $h \leftarrow \mathcal{H}$ raised to all the same exponents. (Note that in the generic group model, this is hard if the above symmetric group assumption is hard.)

Theorem C.4. *The asymmetric q-parallel BDHE assumption implies the (q, q)-*q-ratio *assumption.*

Proof. Suppose we have an adversary $\mathcal{A}$ for the (q,q)-q-ratio assumption. Then we construct an adversary $\mathcal{B}$ for the q-parallel BDHE assumption as follows. $\mathcal{B}$ receives (D,T) and proceeds as follows:

It will implicitly set the generators in the q-ratio assumption to be $(\bar{g},\bar{h}) = (g, h^{dfc^{q+1}})$. It will also choose random $w_1,\ldots,w_q \leftarrow \mathbb{Z}_N$ and implicitly set $u_0 = f$, $u_j = dfb_j$, $v_i = \frac{w_i c^{i-q-1}}{df}$ for $i,j \in [q]$. Finally, it sends $\mathcal{A}$ a challenge consisting of

$$g^f, \{g^{dfb_j}\}_{j\in[q]}, \{(g^{dfc^{q+1-k}b_i/b_j})^{1/w_k}\}_{i,j\in[q],i\neq j,k\in[q]};$$

$$\{(h^{c^i})^{w_i}\}_{i\in[q]}, \{(h^{c^{q+1+i-j}/b_k})^{w_i/w_j}\}_{i,j\in[q],i\neq j,k\in[q]};$$

and

$$T := h^{dc^{q+1}}$$

It then outputs whatever $\mathcal{A}$ does.

To see that this is a perfect simulation of the adversary's view in the q-ratio game, consider the following:

$$g^f = \bar{g}^{u_0}$$

$$g^{dfb_j} = \bar{g}^{u_j}$$

$$(g^{dfc^{q+1-k}b_i/b_j})^{1/w_k} = g^{(\frac{w_k c^{k-q-1}}{df})^{-1} b_i/b_j} = \bar{g}^{\frac{u_i}{u_j v_k}}$$

$$(h^{c^i})^{w_i} = (h^{dfc^{q+1}})^{\frac{w_i c^{i-q-1}}{df}} = \bar{h}^{v_i}$$

$$(h^{c^{q+1+i-j}/b_k})^{w_i/w_j} = (h^{dfc^{q+1}})^{\frac{w_i}{w_j} c^{i-j} \cdot \frac{1}{dfb_k}} = \bar{h}^{\frac{v_i}{v_j u_k}}$$

Finally,

$$h^{dc^{q+1}} = (h^{dfc^{q+1}})^{1/f} = \bar{h}^{1/u_0},$$

so if $\mathcal{B}$ is given a challenge with $h^{dc^{q+1}}$, $\mathcal{A}$'s view will be identical to that in q-ratio when he is given T_0, and if $\mathcal{B}$'s challenge is random, $\mathcal{A}$'s view will be identical to that in q-ratio when he is given T_1. □

D Proof of Security: Remaining Details

Hybrid structure: The following hybrids are defined for $\varphi \in [1,\xi]$ (fix any $b \in \{0,1\}$).

- Hyb_0: This is the real security game $\mathsf{IND\text{-}CPA}^b_{\mathcal{A}}(\lambda, \mathsf{par})$ described in Section 2.1.
- Hyb_1: This game is same as the above except that the ciphertext is semi-functional.
- $\mathsf{Hyb}_{2,\varphi,1}$: This game is same as the above except that $\varphi - 1$ keys are semi-functional, φth key is pseudo-normal, and rest of the keys are normal.
- $\mathsf{Hyb}_{2,\varphi,2}$: This game is same as the above except that the ciphertext is ext-semi-functional and φth key is ext-pseudo-normal.
- $\mathsf{Hyb}_{2,\varphi,3}$: This game is same as the above except that the φth key is ext-pseudo-semi-functional.

- $\mathsf{Hyb}_{2,\varphi,4}$: This game is same as the above except that the ciphertext is semi-functional and φth key is pseudo-semi-functional.
- $\mathsf{Hyb}_{2,\varphi,5}$: This game is same as the above except that the φth key is semi-functional.
- Hyb_3: This game is same as $\mathsf{Hyb}_{2,\xi,5}$ except that the ciphertext is a semi-functional encryption of a random message in $\mathbb{G}_T$.

Our goal is to show that Hyb_0 and Hyb_3 are computationally indistinguishable from each other, irrespective of the bit b used by Chal in the security game $\mathsf{IND\text{-}CPA}^b_{\mathcal{A}}(\lambda, \mathsf{par})$, which implies that Π_P is a secure encryption scheme. First, Hyb_0 and Hyb_1 are shown to be computationally indistinguishable due to the left subgroup indistinguishability property of DSG. After that, the keys are taken one by one from normal to semi-functional space, starting from the first one, by going through five hybrids.

$\mathsf{Hyb}_{2,0,5}$ (or, equivalently, Hyb_1) is computationally indistinguishable from $\mathsf{Hyb}_{2,1,1}$ using the right subgroup indistinguishability property of DSG. Further, $\mathsf{Hyb}_{2,1,1}$ and $\mathsf{Hyb}_{2,1,2}$ are identical in the view of any adversary due to the parameter-hiding property. The transition from $\mathsf{Hyb}_{2,1,2}$ to $\mathsf{Hyb}_{2,1,3}$ is the most complex one involving the new assumption on DSG as well as the symbolic property of PES. The proof is detailed in Lemma 7.3. Once we are in $\mathsf{Hyb}_{2,1,3}$, we can move to $\mathsf{Hyb}_{2,1,4}$ in (almost) the same way as we go from $\mathsf{Hyb}_{2,1,1}$ to $\mathsf{Hyb}_{2,1,2}$, using parameter-hiding. Also, the computational indistinguishability of $\mathsf{Hyb}_{2,1,4}$ and $\mathsf{Hyb}_{2,1,5}$ is very similar to that of $\mathsf{Hyb}_{2,0,3}$ and $\mathsf{Hyb}_{2,1,1}$.

The first key is now in the semi-functional space. The steps above are repeated to transform the other keys as well, till we are in the hybrid $\mathsf{Hyb}_{2,\xi,5}$. The last step of the proof is to show that $\mathsf{Hyb}_{2,\xi,5}$ and Hyb_3 are statistically close to each other.

E New Schemes

Due to the abstract nature of pair encoding framework and the simplicity of our new symbolic property, we have obtained several very general transformations. This leads to several new results. In particular, we have:

- *Constant-size ciphertexts/keys for regular languages.* In Appendix B.5, we proved symbolic security for a pair encoding scheme for regular languages. In this scheme, ciphertext and keys correspond to strings and DFAs, respectively. We can apply the dual-predicate conversion in Section 6.3 to obtain a pair encoding scheme for the dual predicate, where ciphertexts correspond to DFAs. Then, by applying the two transformations in Section 6.1 and 6.2, we get a pair encoding with *a single variable and polynomial* in the ciphertext encoding (but with a bound on the size of DFAs). Both these transformations preserve symbolic property. Finally, by applying the augmentation procedure in Section 5.1 (that adds a few variables and polynomials only), and then using the resulting encoding in the generic transformation (Gen-Trans) of Section 5.4, we get a fully secure encryption scheme in dual system groups under the $\mathsf{q\text{-}ratio}_{\mathsf{dsg}}$ assumption (Theorem 7.1). This leads to an encryption scheme with constant-size ciphertexts in prime-order groups under the k-linear and q-ratio assumptions (Corollary 7.2). The latter assumption is implied by several existing assumptions, see Appendix C.2.

  It is also easy to see that if we don't apply the dual-predicate conversion and apply the transformations for compact keys followed by augmentation and Gen-Trans, we get a fully secure encryption scheme for regular languages with constant-size keys (where keys correspond to bounded-size DFAs).

- *Constant-size ciphertext/keys for doubly spatial encryption.* Even though we don't study doubly spatial encryption in this paper, using the fact that the encoding scheme given by Attrapadung [Att14a] is not trivially broken, and then applying Theorem 4.2, we obtain symbolic property for it.[18] Therefore, we can get fully secure encryption schemes with constant-size ciphertext or keys by applying the transformations in Section 6, augmentation in Section 5.1, and finally Gen-Trans in Section 5.4.

[18] Interestingly, the symbolic security in this case would rely on a computational assumption rather than being shown unconditionally.