Dual System Encryption via Predicate Encodings

Hoeteck Wee[⋆]

ENS, Paris, France

Abstract. We introduce the notion of *predicate encodings*, an information-theoretic primitive reminiscent of linear secret-sharing that in addition, satisfies a novel notion of reusability. Using this notion, we obtain a unifying framework for adaptively-secure public-index predicate encryption schemes for a large class of predicates. Our framework relies on Waters' dual system encryption methodology (Crypto '09), and encompass the identity-based encryption scheme of Lewko and Waters (TCC '10), and the attribute-based encryption scheme of Lewko et al. (Eurocrypt '10). In addition, we obtain several concrete improvements over prior works. Our work offers a novel interpretation of dual system encryption as a methodology for amplifying a one-time private-key primitive (i.e. predicate encodings) into a many-time public-key primitive (i.e. predicate encryption).

[⋆] `wee@di.ens.fr`. CNRS (UMR 8548) and INRIA. Part of this work was done at George Washington University, supported by NSF Awards CNS-1237429 and CNS-1319021.

1 Introduction

Predicate encryption [41, 10, 31] is a new paradigm for public-key encryption that enables fine-grained access control for encrypted data. In predicate encryption, ciphertexts are associated with descriptive values x in addition to a plaintext, secret keys are associated with functions f, and a secret key decrypts the ciphertext if and only if $f(x) = 1$. Here, f may express an arbitrarily complex access policy, which is in stark contrast to traditional public-key encryption, where access is all or nothing. Predicate encryption generalizes both identity-based encryption (IBE) [42, 8, 18] where f checks for equality, and attribute-based encryption (ABE) [41, 26], where f encodes a boolean formula. The security requirement for predicate encryption enforces resilience to collusion attacks, namely any group of users holding secret keys for different functions learns nothing about the plaintext if none of them is individually authorized to decrypt the ciphertext. This should hold even if the adversary *adaptively* decides which secret keys to ask for.

Terminology. Throughout this work, we use *predicate encryption* to refer to public-index predicate encryption, and reserve attribute-based encryption for the special case where the predicate is computed by a boolean formula.

Dual system encryption. In [44], Waters introduced the powerful *dual system encryption* methodology for building adaptively secure predicate encryption. In a dual system encryption scheme, there are two types of keys and ciphertexts: normal and semi-functional. Normal keys and ciphertexts are used in the real system, while the semi-functional objects are gradually introduced in the hybrid security proof. The proofs are often quite complex and delicate. In spite of the large body of work relying on the dual system encryption methodology (e.g. [33, 36, 39, 34, 37, 32, 40]), there seems to be no concrete, overarching framework explaining these schemes. In particular, even in the simplest information-theoretic setting in composite-order groups, we do not have a clear understanding of why the Lewko-Waters heuristic [33] for deriving dual system encryption schemes via "embedding" works for the IBE scheme in [7] but not the ABE scheme in [26] (even under the "one use" restriction). We also do not have a formal, systematic approach for deriving the semi-functional objects used in the security proof: for instance, the semi-functional keys in the dual system IBE in [33] have independent random semi-functional components, whereas those in the ABE scheme in [36] are carefully designed to have certain correlations.

Decoupling functionalities? A recurring trend in predicate encryption, which arose with both the introduction of dual system encryption and lattice-based techniques [23, 13], is a systematic adaption of prior selectively secure in bilinear groups to achieve either improved parameters (e.g. shorter ciphertexts for HIBE) or larger classes of functionalities (e.g. from IBE to ABE). Moreover, the new schemes often bear a structural resemblance to prior schemes. The phenomenon suggests that we should aim to decouple the way we encode a predicate/functionality in an encryption scheme from the design and analysis of the scheme.

1.1 Our contributions

We present a framework for the design and analysis of dual system encryption schemes in composite-order bilinear groups, which allows us to also decouple the predicate from the security proof. The crux of our framework is a notion of *predicate encodings*. Roughly speaking, predicate encodings are an information-theoretic primitive reminiscent of secret-sharing schemes that in addition, satisfies a novel notion of reusability. Using predicate encodings, we obtain new insights into the dual system encryption methodology and new concrete predicate encryption schemes. Before we describe our results, we present an overview of predicate encodings.

Predicate encodings. A predicate encoding for a Boolean predicate $\mathsf{P}(\cdot,\cdot)$, is specified by a pair of algorithms $(\mathsf{sE},\mathsf{rE})$ with a common private input w and in addition,

- sender encoding sE takes as input (x,w) and outputs $\mathsf{sE}(x,w)$.
- receiver encoding rE takes as input (α,y,w) and randomness r, and outputs $\mathsf{rE}(\alpha,y,w;r)$.

The basic requirements for α are the same as that for secret-sharing:

(reconstruction.) if $\mathsf{P}(x,y)=1$, we can recover α from the encodings;

(privacy.) if $\mathsf{P}(x,y)=0$, the encodings hide α perfectly.

The key conceptual novelty in predicate encoding (over other existing notions e.g. [45, 4, 20, 28, 24, 29, 1]) which enables us to handle collusions in predicate encryption is w**-hiding**. Informally, w-hiding stipulates that we can hide all information about w in the receiver encoding by setting the randomness r to some fixed value (e.g. we can hide w in the expression rw by setting r to 0). Note that the definition of w-hiding treat w and r differently. Finally, we impose some algebraic structure in the encodings similar to that for linear secret-sharing, in order to carry out encoding and reconstruction "in the exponent" in the encryption scheme.

We stress that the requirements for predicate encodings are fairly basic and indeed, we readily obtain predicate encodings for a large class of predicates like HIBE, doubly spatial encryption and ABE, many of which are implicit in prior selectively secure schemes [7, 11, 9]. Moreover, privacy for these encodings follows readily from linear algebra, as is typically the case for information-theoretic primitives and constructions. On the other hand, the encodings in [26, 3] do not satisfying our requirements (c.f. Section 5.5); this provides a partial explanation as to why the Lewko-Waters heuristic [33] cannot be applied to these schemes.

Predicate encryption from predicate encodings. Starting from a predicate encoding for P, we construct a predicate encryption scheme in composite-order bilinear groups whose order is the product of three primes p_1,p_2,p_3, and establish adaptive security in a *modular* manner via Waters' dual system encryption methodology. Here, a secret key sk_y can decrypt a ciphertext ct_x iff $\mathsf{P}(x,y)=1$. We associate ciphertext with sender encoding and secret keys with receiver encodings. Correctness will rely on the reconstruction property modulo p_1, whereas security against collusions will rely on privacy and w-hiding modulo p_2. The third subgroup corresponding to p_3 is used for additional randomization which we ignore in this overview. Roughly speaking, the master public key, secret key and ciphertext are of the form:

$$\mathsf{mpk} := (g_1, g_1^w, e(g_1,g_1)^\alpha), \qquad \mathsf{sk}_y := g_1^{\mathsf{rE}(\alpha,y,w;r)}, \qquad \mathsf{ct}_x := ((g_1^{\mathsf{sE}(x,w)})^s, e(g_1,g_1)^{\alpha s}\cdot m)$$

where g_1 is a generator of order p_1. Observe that the lengths of w, sE and rE correspond naturally to the sizes of the public parameters, ciphertexts and secret keys. If $\mathsf{P}(x, y) = 1$, decryption works by reconstructing α from $\mathsf{sE}(x, w)$ and $\mathsf{rE}(\alpha, y, w; r)$ in the exponent via a pairing.

Proof strategy. We outline the key challenges in establishing adaptive security of the predicate encryption scheme, which yields new insights into dual system encryption methodology:

- First, predicate encoding is essentially a private-key primitive, in that α-privacy against an adversary that does not see the shared randomness w, whereas w must be made public in order that encryption uses the same w as that used for decryption. The scheme overcomes this conundrum by publishing only g_1^w in the public parameters. This leaks information about $w \pmod{p_1}$ so that we can exploit α-reconstruction modulo p_1, but completely hides $w \pmod{p_2}$ so that α-privacy holds modulo p_2. In the final step in the hybrid security proof, the message is masked by α modulo p_2 whereas the public parameters and all secret keys reveal no information about α modulo p_2. Security then follows via a simple information-theoretic argument.
- Second, predicate encoding only provides one-time security, that is, α-privacy no longer holds if we use w across more than one receiver encoding, as will be the case when an adversary requests multiple secret keys. We overcome this difficulty by ensuring that in each step in the proof of security, at most one secret key leaks information about $w \pmod{p_2}$. In particular, both normal and semi-functional keys reveals no information about $w \pmod{p_2}$. We only leak information about $w \pmod{p_2}$ when transitioning from a normal to a semi-functional key, one key at a time. During the transition, we rely on w-hiding to "erase" information about $w \pmod{p_2}$ from all remaining keys (see Fig 2 and Lemma 3).
- Finally, predicate encoding only provides non-adaptive security, namely α-privacy only holds if x, y are fixed in advance. On the other hand, an adversary may choose a key query y after seeing the challenge ciphertext for x, which leaks $\mathsf{rE}(x, w, r)$. This is where we rely crucially on the fact that the encoding achieves *perfect* α-privacy, for which non-adaptive implies adaptive privacy. To the best of our knowledge, this is the first time this requirement is explicitly pointed out for use in dual system encryption. (A recent work [6] highlights several subtleties in defining and achieving adaptive privacy in the related setting of garbled circuits.)

In short, dual system encryption allows us to boost security in a private-key, one-time, non-adaptive setting to a full-fledged public-key, many-time, adaptive setting! Along the way, we introduce a conceptual simplification where we define the semi-functional entities via auxiliary algorithms, reminiscent of Cramer-Shoup projective hashing [19].

Instantiations. Our final predicate encryption scheme is adaptively secure under the standard Subgroup Decision Assumptions in composite order bilinear groups. We note that our implementation of the dual system encryption methodology differs in subtle ways from prior composite-order instantiations in [33, 36] (see e.g. Remark 2). In addition to a unifying proof of security for a large class of predicates, we obtain the several concrete improvements over prior works:

- We eliminate the need for an additional computational assumption which refers to the target group, as used in the prior composite-order HIBE and ABE [33, 36]. In particular, we show how to execute the final transition in the proof of security with an information-theoretic argument instead of a computational one.

- We reduce the key size of the (key-policy) ABE in [36] by half. The improvement comes from eliminating some redundant randomization in the associated encoding.
- We obtain novel (to the best of our knowledge) and simple constructions of adaptively-secure non-zero inner product encryption and doubly spatial encryption in composite-order bilinear groups.

1.2 Discussion

Predicate encodings decouple and modularize the essential information-theoretic properties from the broader mechanics of a dual system cryptosystem and its analysis. Previous dual-system proofs are often monolithic and hard to follow, and the core new ideas are sometimes buried underneath lots of algebraic notation that is repeated (or only slightly tweaked) from one scheme to another. Our framework allows us to distill the core argument that is common to dual system cryptosystems from a separate information-theoretic argument which is tailored to the underlying predicate.

Open problems. This work raises a number of open problems.

- Do bilinear (or multi-linear) predicate encodings exist for all polynomial-time computable predicates? An affirmative answer would yield adaptively secure ABE for circuits [25, 21], without relying on complexity leveraging. However, even achieving perfect α-hiding without the bilinear requirements would likely require overcoming long-standing barriers.
- Can we prove lower bounds on the length of $\mathbf{w}$, rE or sE for predicate encodings (corresponding to public parameters, secret keys and ciphertext sizes respectively)? In particular, the encodings for ABE require that rE grows with the size of the formula (c.f. Section 5.5) and we conjecture that such a dependency is in fact necessary for perfect α-privacy.
- Finally, we note that our work does not cover more recent applications of dual system encryption in the computational setting for ABE with short ciphertexts [35]. There, α-privacy is computational, for which we no longer get adaptive from non-adaptive security "for free". We leave these extensions for future work.

Subsequent work. In subsequent works [15, 16], we built upon the ideas introduced here in several ways. In [15], we introduced *dual system groups*, a step towards abstracting the underlying group structure needed to support the dual system encryption methodology. This is orthogonal and complementary to this work, which is about abstracting how we encode the predicate/functionality. In [16], we presented the first adaptively secure IBE where the security loss does not depend on the number of secret key queries, partially resolving an open problem in [43, 22]. The crucial insight lies in replacing the one-time predicate encoding for IBE (a randomized MAC) with a reusable one (a pseudorandom function). Specifically, we rely on dual system encryption methodology to "compile" the Naor-Reingold PRF [38] which is a private-key primitive into a fully secure IBE.

Organization. We formalize predicate encodings in Section 3. We present the generic construction of a predicate encryption scheme in Section 4. We describe instantiations of predicate encodings in Section 5. Preliminaries are given in Section 2.

2 Preliminaries

Notation. We denote by $s \leftarrow_{\mathrm{R}} S$ the fact that s is picked uniformly at random from a finite set S. By PPT, we denote a probabilistic polynomial-time algorithm. Throughout, we use 1^λ as the security parameter. We use $\cdot$ to denote multiplication as well as component-wise multiplication. We use lower case boldface to denote (column) vectors over scalars and upper case boldcase to denote vectors of group elements as well as matrices. Given two vectors $\mathbf{x} = (x_1, x_2, \ldots), \mathbf{y} = (y_1, y_2, \ldots)$ over scalars, we use $\langle \mathbf{x}, \mathbf{y} \rangle$ to denote the standard dot product $\mathbf{x}^\top \mathbf{y}$. Given a group element g, we write $g^{\mathbf{x}}$ to denote $(g^{x_1}, g^{x_2}, \ldots)$.

2.1 Composite Order Bilinear Groups and Cryptographic Assumptions

We instantiate our system in composite order bilinear groups, which were introduced in [12] and used in [31, 33, 36]. A generator $\mathcal{G}$ takes as input a security parameter λ and outputs a description $\mathbb{G} := (N, G, G_T, e)$, where N is product of distinct primes of $\Theta(\lambda)$ bits, G and G_T are cyclic groups of order N, and $e : G \times G \to G_T$ is a non-degenerate bilinear map. We require that the group operations in G and G_T as well the bilinear map e are computable in deterministic polynomial time.We consider bilinear groups G whose orders N are products of three distinct primes p_1, p_2, p_3 (that is, $N = p_1 p_2 p_3$). We can write $G = G_{p_1} G_{p_2} G_{p_3}$ where $G_{p_1}, G_{p_2}, G_{p_3}$ are subgroups of G of order p_1, p_2 and p_3 respectively. In addition, we use $G^*_{p_i}$ to denote $G_{p_i} \setminus \{1\}$. We will often write g_1, g_2, g_3 to denote random generators for the subgroups $G_{p_1}, G_{p_2}, G_{p_3}$ of order p_1, p_2 and p_3 respectively.

Cryptographic assumptions. Our construction relies on the following two assumptions which are essentially the first two of three assumptions used in [33, 36] and are instances of the General Subgroup Decision Assumption in composite-order groups [5]. We define the following two advantage functions:

$$\begin{aligned}
&\mathsf{Adv}^{\mathrm{SD1}}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\mathbb{G}, D, T_0) = 1] - \Pr[\mathcal{A}(\mathbb{G}, D, T_1) = 1]| \\
&\quad \text{where } \mathbb{G} \leftarrow \mathcal{G}, T_0 \leftarrow \boxed{G_{p_1}}, T_1 \leftarrow_{\mathrm{R}} \boxed{G_{p_1} G_{p_2}} \\
&\quad\quad \text{and } D := (g_1, g_3, g_{\{1,2\}}), g_1 \leftarrow_{\mathrm{R}} G^*_{p_1}, g_3 \leftarrow_{\mathrm{R}} G^*_{p_3}, g_{\{1,2\}} \leftarrow_{\mathrm{R}} G_{p_1} G_{p_2} \\
&\mathsf{Adv}^{\mathrm{SD2}}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\mathbb{G}, D, T_0) = 1] - \Pr[\mathcal{A}(\mathbb{G}, D, T_1) = 1]| \\
&\quad \text{where } \mathbb{G} \leftarrow \mathcal{G}, T_0 \leftarrow \boxed{G^*_{p_1} G_{p_3}}, T_1 \leftarrow_{\mathrm{R}} \boxed{G^*_{p_1} G^*_{p_2} G_{p_3}} \\
&\quad\quad \text{and } D := (g_1, g_3, g_{\{1,2\}}, g_{\{2,3\}}), g_1 \leftarrow_{\mathrm{R}} G^*_{p_1}, g_3 \leftarrow_{\mathrm{R}} G^*_{p_3}, g_{\{1,2\}} \leftarrow_{\mathrm{R}} G_{p_1} G_{p_2}, g_{\{2,3\}} \leftarrow_{\mathrm{R}} G_{p_2} G_{p_3}
\end{aligned}$$

Assumption 1 (resp. 2) asserts that for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathrm{SD1}}_{\mathcal{G},\mathcal{A}}(\lambda)$ (resp. $\mathsf{Adv}^{\mathrm{SD2}}_{\mathcal{G},\mathcal{A}}(\lambda)$) is a negligible function in λ.

2.2 Predicate Encryption

We define predicate encryption in the framework of key encapsulation. A predicate encryption scheme for a predicate $\mathsf{P}(\cdot, \cdot)$ consists of four algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}) \to (\mathsf{pp}, \mathsf{mpk}, \mathsf{msk})$. The setup algorithm gets as input the security parameter λ, the attribute universe $\mathcal{X}$, the predicate universe $\mathcal{Y}$ and outputs the public parameter $(\mathsf{pp}, \mathsf{mpk})$, and the master key msk. All the other algorithms get pp as part of its input.

$\mathsf{Enc}(\mathsf{mpk}, x) \to (\mathsf{ct}_x, \kappa)$. The encryption algorithm gets as input mpk and an attribute $x \in \mathcal{X}$. It outputs a ciphertext ct_x and a symmetric key $\kappa \in \{0,1\}^\lambda$. Note that x is public given ct_x.

$\mathsf{KeyGen}(\mathsf{msk}, y) \to \mathsf{sk}_y$. The key generation algorithm gets as input msk and a value $y \in \mathcal{Y}$. It outputs a secret key sk_y. Note that y is public given sk_y.

$\mathsf{Dec}(\mathsf{sk}_y, \mathsf{ct}_x) \to \kappa$. The decryption algorithm gets as input sk_y and ct_x such that $\mathsf{P}(x, y) = 1$. It outputs a symmetric key κ.

Correctness. We require that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 1$,

$$\Pr[(\mathsf{ct}_x, \kappa) \leftarrow \mathsf{Enc}(\mathsf{mpk}, x); \mathsf{Dec}(\mathsf{sk}_y, \mathsf{ct}_x) = \kappa)] = 1,$$

where the probability is taken over $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y})$ and the coins of Enc.

Security definition. For a stateful adversary $\mathcal{A}$, we define the advantage function

$$\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda) := \Pr\left[b = b' : \begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}); \\ x \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk}, \cdot)}(\mathsf{mpk}); \\ b \leftarrow_{\mathrm{R}} \{0,1\}; \kappa_1 \leftarrow_{\mathrm{R}} \{0,1\}^\lambda \\ (\mathsf{ct}_x, \kappa_0) \leftarrow \mathsf{Enc}(\mathsf{mpk}, x); \\ b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk}, \cdot)}(\mathsf{ct}_x, \kappa_b) \end{array} \right] - \frac{1}{2}$$

with the restriction that all queries y that $\mathcal{A}$ makes to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ satisfies $\mathsf{P}(x, y) = 0$ (that is, sk_y does not decrypt ct_x). A predicate encryption scheme is *adaptively secure* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ.

3 Bilinear Predicate Encodings

In this section, we describe *predicate encodings* more formally. Then, we discuss several examples, before describing the *bilinear* requirement.

3.1 Predicate encodings

Fix a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. A *predicate encoding* for P is a pair of algorithms $(\mathsf{sE}, \mathsf{rE})$, where sE is deterministic and takes as input $(x, w) \in \mathcal{X} \times \mathcal{W}$; and rE is randomized and takes as input $(\alpha, y, w) \in \mathcal{D} \times \mathcal{Y} \times \mathcal{W}$ and randomness $r \in \mathcal{R}$. (We stress that $\mathcal{W}$ and $\mathcal{R}$ play very different roles, as evident in the w-hiding property.) In addition, we require that $(\mathsf{sE}, \mathsf{rE})$ satisfy the following three properties:

(α-reconstruction.) For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 1$ and for all r, we can (efficiently) recover α given $x, y, \mathsf{sE}(x, w), \mathsf{rE}(\alpha, y, w; r)$.

(α-privacy.) For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 0$, and for all $\alpha \in \mathcal{D}$, the joint distribution $\mathsf{sE}(x, w), \mathsf{rE}(\alpha, y, w; r)$ *perfectly* hides α. That is, for all $\alpha, \alpha' \in \mathcal{D}$, the following joint distributions are *identically* distributed:

$$\{x, y, \alpha, \mathsf{sE}(x, w), \mathsf{rE}(\alpha, y, w; r)\} \quad \text{and} \quad \{x, y, \alpha, \mathsf{sE}(x, w), \mathsf{rE}(\alpha', y, w; r)\}$$

where the randomness is taken over $(w, r) \leftarrow_{\mathrm{R}} \mathcal{W} \times \mathcal{R}$.

(w-hiding.) There exists some element $0 \in \mathcal{R}$ such that for all $(\alpha, y, w) \in \mathcal{D}\times \in \mathcal{Y} \times \mathcal{W}$, $\mathsf{rE}(\alpha, y, w; 0)$ is statistically independent of w, that is, for all $w' \in \mathcal{W}$:

$$\mathsf{rE}(\alpha, y, w; 0) = \mathsf{rE}(\alpha, y, w'; 0)$$

Remark 1. We rely crucially on the fact that α is *perfectly* hidden in the proof of security, so that non-adaptive indistinguishability implies adaptive indistinguishability. (This is not true in the statistical or computational setting.) Concretely, we claim that α-privacy implies that even if y is chosen *adaptively* after seeing $(x, \alpha, \mathsf{rE}(x, w))$, the distributions

$$\mathsf{rE}(\alpha, y, w; r) \quad \text{and} \quad \mathsf{rE}(0, y, w; r)$$

are perfectly indistinguishable. This simply follows from the fact that an adaptive distinguisher with advantage ϵ can be converted into a non-adaptive distinguisher with advantage $\epsilon/|\mathcal{Y}|$ via random guessing. Since any non-adaptive distinguisher has advantage 0, we must have $\epsilon = 0$ to begin with. The same argument applies to the setting where x is chosen *adaptively* after seeing $(y, \alpha, \mathsf{rE}(\cdot, y, w; r))$.

Remark 2. We note that w-hiding as defined is not the only way to achieve "w-reusability". For instance, for the equality predicate as in IBE, the Lewko-Waters scheme [33] achieves reusability by essentially masking $\mathsf{rE}(\alpha, y, w; r)$ with a fresh one-time pad for each secret key query. This works for IBE and HIBE because $\mathsf{rE}(\alpha, y, w; r)$ has the uniform distribution for every y. However, this approach does not work for the ABE predicate. Indeed, by using w-hiding, we obtain a different proof of security of the Lewko-Waters HIBE.

Example 1: equality. Fix an integer N to be the product of three λ-bit primes. Consider the equality predicate where $\mathcal{X} = \mathcal{Y} = [N]$ and $\mathsf{P}(x, y) = 1$ iff $x = y$. The following is a predicate encoding for equality used in [7, 33]:

- $\mathcal{D} := \mathbb{Z}_N; \mathcal{W} := \mathbb{Z}_N^2; \mathcal{R} := \mathbb{Z}_N^*$.
- $\mathsf{sE}(x, (w_1, w_2)) := (1, w_1 + w_2 x)$
- $\mathsf{rE}(\alpha, y, (w_1, w_2); r) := (\alpha + r(w_1 + w_2 y), -r)$

For α-reconstruction when $x = y$, simply take the dot product of the two vectors. For α-privacy when $x \neq y$,[1] we exploit the fact that $(w_1 + w_2 x, w_1 + w_2 y)$ are pairwise independent and $r \in \mathbb{Z}_N^*$. (Note that perfect α-privacy does not hold if we set $\mathcal{R}$ to be $\mathbb{Z}_N$ instead of $\mathbb{Z}_N^*$.) To achieve w-hiding, we simply set $r = 0$.[2]

Example 2: equality. Consider the same predicate and construction as before, but replace rE by

$$\mathsf{rE}(\alpha, y, (w_1, w_2)) := (\alpha + (w_1 + w_2 y), -1).$$

[1] Here, we will even assume $\gcd(x - y, N) = 1$; otherwise, we can find a non-trivial factor of N.

[2] This does not actually work since $0 \notin \mathcal{R}$, but we will consider a slight weakening of w-hiding in the next section.

This still satisfies α-reconstruction and α-privacy, but not $\mathbf{w}$-hiding nor linear receiver encoding (the latter property is defined in the next Section).

3.2 Bilinearity

Fix a prime p. Let $(\mathsf{sE}, \mathsf{rE})$ be a predicate encoding for $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, where $\mathcal{X}$ and $\mathcal{Y}$ may depend on p. We say that $(\mathsf{sE}, \mathsf{rE})$ is *p-bilinear* if it satisfies the following properties:

(input domains.) $\mathcal{D} = \mathbb{Z}_p$, $\mathcal{W} = \mathbb{Z}_p^{\ell_{\mathcal{W}}}$ and $\mathcal{R} = \mathbb{Z}_p^{\ell_{\mathcal{R}}} \times (\mathbb{Z}_p^*)^{\ell'_{\mathcal{R}}}$ for some integers $\ell_{\mathcal{W}}, \ell_{\mathcal{R}}, \ell'_{\mathcal{R}}$.[3]

(output domains.) The output of sE and rE are (column) vectors over $\mathbb{Z}_p$.

(affine sender encoding.) For all $x \in \mathcal{X}$, $\mathsf{sE}(x, \cdot)$ is affine in $\mathbf{w}$.

(linear receiver encoding.) For all $(\alpha, y, \mathbf{w}) \in \mathcal{D} \times \mathcal{Y} \times \mathcal{W}$, $\mathsf{rE}(\cdot, y, \mathbf{w}; \cdot)$ is linear in $\alpha, \mathbf{r}$.

(bilinear α-reconstruction.) For all (x, y) such that $\mathsf{P}(x, y) = 1$, we can efficiently compute a linear map $\mathbf{M}_{xy}$ (a matrix over $\mathbb{Z}_p$) such that for all $\mathbf{r} \in \mathcal{R}$,

$$\mathsf{sE}(x, \mathbf{w})^\top \mathbf{M}_{xy} \mathsf{rE}(\alpha, y, \mathbf{w}; \mathbf{r}) = \alpha$$

(w-hiding.) For all $(\alpha, y, \mathbf{w}) \in \mathcal{D} \times \in \mathcal{Y} \times \mathcal{W}$, we have

$$\mathsf{rE}(\alpha, y, \mathbf{w}; \mathbf{0}) = \mathsf{rE}(\alpha, y, \mathbf{0}; \mathbf{0})$$

where we use $\mathbf{0}$ to refer to the all zeroes vector in $\mathbb{Z}_p^{\ell_{\mathcal{W}}}$ and in $\mathbb{Z}_p^{\ell_{\mathcal{R}} + \ell_{\mathcal{R}'}}$.[4]

The above definition extends to any integer N by replacing $\mathbb{Z}_p, \mathbb{Z}_p^*$ with $\mathbb{Z}_N, \mathbb{Z}_N^*$ respectively.

Remark 3. We will exploit the affine sender encoding and linear receiver encoding to compute sE and rE "in the exponent". Fix $g \in G_N$.

- Affine sender encoding implies that given $x \in \mathcal{X}$ along with $g, g^{\mathbf{w}}$, we can compute $g^{\mathsf{sE}(x, \mathbf{w})}$; indeed, we will slightly abuse notation and write this as $\mathsf{sE}(x, g^{\mathbf{w}})$.
- Similarly, linear receiver encoding implies that given $(y, \mathbf{w}) \in \mathcal{Y} \times \mathcal{W}$ along with $g^\alpha, g^{\mathbf{r}}$ (but not g), we can compute $g^{\mathsf{rE}(\alpha, y, \mathbf{w}; \mathbf{r})}$; again, we will write this as $\mathsf{rE}(g^\alpha, y, \mathbf{w}; g^{\mathbf{r}})$.

Extensions. We also consider two extensions, first to handle randomized sender's encoding in Section 5.5 and second to support delegation in Section 5.3.

4 Predicate Encryption from Bilinear Encoding

We present a predicate encryption scheme in composite-order bilinear groups whose order is the product of three primes (c.f. Section 2.1), for any predicate $\mathsf{P}(\cdot, \cdot)$ which admits a bilinear predicate encoding. In addition, we show that the scheme is adaptively secure under the General Subgroup Decision Assumption. We refer to Section 1.1 for an overview of the construction and the proof.

[3] The distinction between $\mathbb{Z}_p$ and $\mathbb{Z}_p^*$ is significant because we require *perfect* α-privacy.

[4] This is in fact a slight relaxation of the general w-hiding property since $\mathbf{0}$ does not lie in $\mathcal{R}$ whenever $\ell'_R > 0$.

Property	Where it is used
bilinear α-reconstruction	Dec and correctness
affine sender encoding	Enc
linear receiver encoding	$\mathsf{KeyGen}, \widehat{\mathsf{KeyGen}}$
α-privacy	pseudo-normal to pseudo-SF secret keys, Lemma 3
$\mathbf{w}$-hiding	pseudo-normal to pseudo-SF secret keys, Lemma 3

Fig. 1. Properties of predicate encodings and where they are used

4.1 Construction

Fix a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. Given a N-bilinear predicate encoding $(\mathsf{sE}, \mathsf{rE})$ for P, we may construct a predicate encryption scheme for P as follows:

$\mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y})$: On input $(1^\lambda, \mathcal{X}, \mathcal{Y})$, first generate $\mathbb{G} \leftarrow \mathcal{G}(1^\lambda)$, then sample $\mathsf{H} : G_T \to \{0,1\}^\lambda$ from a family of pairwise-independent hash functions. In addition, sample $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_N, \mathbf{w} \leftarrow_{\mathrm{R}} \mathcal{W}$, and output[5]

$$\mathsf{pp} := (\mathbb{G}, \mathsf{H}, g_1, g_3) \quad \text{and} \quad \mathsf{mpk} := (g_1^{\mathbf{w}}, e(g_1, g_1)^\alpha) \quad \text{and} \quad \mathsf{msk} := (g_1^\alpha g_2^\alpha, \mathbf{w})$$

$\mathsf{KeyGen}(\mathsf{msk}, y)$: On input $\mathsf{msk} = (g_1^\alpha g_2^\alpha, \mathbf{w})$ and a predicate y, sample $\mathbf{r} \leftarrow_{\mathrm{R}} \mathcal{R}$ and output[6]

$$\mathsf{sk}_y := \mathsf{rand3}(\mathsf{rE}(g_1^\alpha g_2^\alpha, y, \mathbf{w}; g_1^{\mathbf{r}})) = \mathsf{rand3}(g_1^{\mathsf{rE}(\alpha, y, \mathbf{w}; \mathbf{r})} \cdot g_2^{\mathsf{rE}(\alpha, y, \mathbf{w}; \mathbf{0})})$$

Here, $\mathsf{rand3}$ is an algorithm that randomizes the G_{p_3}-components, namely on input a vector $\mathbf{C} \in G_N^\ell$, outputs $\mathbf{C} \cdot g_3^{\mathbf{r}'}$ where $\mathbf{r}' \leftarrow_{\mathrm{R}} \mathbb{Z}_N^\ell$.

$\mathsf{Enc}(\mathsf{mpk}, x)$: On input an attribute $x \in \mathcal{X}$, sample $s \leftarrow_{\mathrm{R}} \mathbb{Z}_N$ and output the ciphertext and symmetric key

$$\mathsf{ct}_x := (\mathsf{sE}(x, g_1^{\mathbf{w}}))^s = g_1^{\mathsf{sE}(x, \mathbf{w}) s} \quad \text{and} \quad \kappa := \mathsf{H}((e(g_1, g_1)^\alpha)^s)$$

$\mathsf{Dec}(\mathsf{sk}_y, \mathsf{ct}_x)$: On input sk_y and ct_x where $\mathsf{P}(x, y) = 1$, output

$$\mathsf{H}(e(\mathsf{ct}_x, \mathsf{sk}_y^{\mathbf{M}_{xy}}))$$

where $\mathbf{M}_{xy}$ is the matrix for bilinear reconstruction and $e(\mathsf{ct}_x, \mathsf{sk}_y^{\mathbf{M}_{xy}}) := \sum_i e((\mathsf{ct}_x)_i, \sum_j (\mathsf{sk}_y)_j^{(\mathbf{M}_{xy})_{i,j}})$.

[5] If we want to be able to derive multiple $(\mathsf{mpk}, \mathsf{msk})$ from the same pp, we will need to append a random generator of $G_{\{1,2\}}$ to pp, which we can then use to sample msk. Note that this will not affect the proof of security, since such a generator is provided to the distinguisher in both Assumption 1 and 2.

[6] Refer to Remark 3 for the notation $\mathsf{rE}(\cdots)$ as used here.

Correctness. For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 1$, we have

$$\begin{aligned}\mathsf{Dec}(\mathsf{sk}_y, \mathsf{ct}_x) &= \mathsf{Dec}(g_1^{\mathsf{rE}(\alpha,y,\mathbf{w};\mathbf{r})} g_2^{\mathsf{rE}(\alpha,y,\mathbf{w};\mathbf{0})}\mathbf{Z}_3, g_1^{\mathsf{sE}(x,\mathbf{w})s}) \\ &= \mathsf{H}(e(g_1^{\mathsf{sE}(x,\mathbf{w})s}, (g_1^{\mathsf{rE}(\alpha,y,\mathbf{w};\mathbf{r})} g_2^{\mathsf{rE}(\alpha,y,\mathbf{w};\mathbf{0})}\mathbf{Z}_3)^{\mathbf{M}_{xy}})) \\ &= \mathsf{H}(e(g_1^{\mathsf{sE}(x,\mathbf{w})s}, (g_1^{\mathsf{rE}(\alpha,y,\mathbf{w};\mathbf{r})})^{\mathbf{M}_{xy}})) \\ &= \mathsf{H}(e(g_1, g_1)^{\langle \mathsf{sE}(x,\mathbf{w})s, \mathbf{M}_{xy}\mathsf{rE}(\alpha,y,\mathbf{w};\mathbf{r})\rangle}) \\ &= \mathsf{H}((e(g_1, g_1)^{\alpha})^{s})\end{aligned}$$

4.2 Proof of security

We prove the following theorem:

Theorem 1. *Under Assumptions 1 and 2 (c.f. Section 2.1), the predicate encryption scheme described in Section 4.1 is adaptively secure (c.f. Section 2.2). More precisely, for any adversary $\mathcal{A}$ that makes at most q queries against the predicate encryption scheme, there exist adversaries $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{PE}}(\lambda) \le \mathsf{Adv}_{\mathcal{G},\mathcal{A}_1}^{\mathrm{SD1}}(\lambda) + q \cdot \mathsf{Adv}_{\mathcal{G},\mathcal{A}_2}^{\mathrm{SD2}}(\lambda) + q \cdot \mathsf{Adv}_{\mathcal{G},\mathcal{A}_3}^{\mathrm{SD2}}(\lambda) + 2^{-\Omega(\lambda)}$$

The proof follows via a series of games, outlined in Section 1.1 and summarized in Fig 2. Following Waters' dual system encryption metholodogy [44], there are two types of keys and ciphertexts: normal and semi-functional. We first describe two auxiliary algorithms (analogous to "private evaluation" algorithms in Cramer-Shoup projective hashing [19]), and then defining the semi-functional distributions via these auxiliary algorithms.

Auxiliary algorithms. We consider the following algorithms: a deterministic algorithm $\widehat{\mathsf{Enc}}$ for computing ciphertexts and a randomized algorithm $\widehat{\mathsf{KeyGen}}$ for computing secret keys.

$\widehat{\mathsf{Enc}}(\mathsf{pp}, x; \mathsf{msk}', C)$: On input $x \in \mathcal{X}$, along with $\mathsf{msk}' = (h, \mathbf{w}) \in G_N \times \mathcal{W}$ and $C \in G_N$, output:

$$(\mathsf{ct}_x, \kappa) := (C^{\mathsf{sE}(x,\mathbf{w})}, \mathsf{H}(e(C, h))$$

Observe that for all $(\mathsf{pp}, \mathsf{mpk}, \mathsf{msk})$ output by Setup and for all $s \in \mathbb{Z}_N$, we have

$$\mathsf{Enc}(\mathsf{mpk}, x; s) = \big(g_1^{\mathsf{sE}(x,\mathbf{w})s}, \mathsf{H}(e(g_1, g_1)^{\alpha s})\big) = \widehat{\mathsf{Enc}}(\mathsf{pp}, x; \mathsf{msk}, g_1^s)$$

$\widehat{\mathsf{KeyGen}}(\mathsf{msk}', y; R)$: On input $\mathsf{msk}' = (h, \mathbf{w}) \in G_N \times \mathcal{W}$, $y \in \mathcal{Y}$ and $R \in G_N$, sample $\mathbf{r} \leftarrow_{\mathrm{R}} \mathcal{R}$ and output

$$\mathsf{sk}_y := \mathsf{rand3}(\mathsf{rE}(h, y, \mathbf{w}; R^{\mathbf{r}}))$$

Observe that, for any msk', y and any $R \in G_{p_1}^* G_{p_3}$, the following three distributions are identical:

$$\mathsf{KeyGen}(\mathsf{msk}', y) \quad \text{and} \quad \widehat{\mathsf{KeyGen}}(\mathsf{msk}', y; g_1) \quad \text{and} \quad \widehat{\mathsf{KeyGen}}(\mathsf{msk}', y; R)$$

That is, we have three different but equivalent ways to generate real secret keys. The equivalence of the first two distributions is straight-forward. For the equivalence of the second and the third, we use the fact that $\mathcal{R}$ is of the form $\mathbb{Z}_N^{\ell_{\mathcal{R}}} \times (\mathbb{Z}_N^*)^{\ell'_{\mathcal{R}}}$ and that we randomize using $\mathsf{rand3}$.

Game	Ciphertext / Key (ct_x, κ)	Secret Key sk_y	Justification	Remark
0	$(\mathbf{0}, 0)$	$\mathsf{rE}(\alpha, y, \mathbf{w}; \mathbf{0})$		actual scheme
1	$\boxed{(\mathsf{sE}(x, \mathbf{w})s, \alpha s)}$	$\mathsf{rE}(\alpha, y, \mathbf{w}; \mathbf{0})$	Assumption 1	normal to SF (ct_x, κ)
2.i.1	$(\mathsf{sE}(x, \mathbf{w})s, \alpha s)$	$\mathsf{rE}(\alpha, y, \mathbf{w}; \boxed{\mathbf{r}})$	Assumption 2	normal to pseudo-normal sk_y
3.i.2	$(\mathsf{sE}(x, \mathbf{w})s, \alpha s)$	$\mathsf{rE}(\boxed{0}, y, \mathbf{w}; \mathbf{r})$	α-privacy & $\mathbf{w}$-hiding	pseudo-normal to pseudo-SF sk_y
2.i.3	$(\mathsf{sE}(x, \mathbf{w})s, \alpha s)$	$\mathsf{rE}(0, y, \mathbf{w}; \boxed{\mathbf{0}})$	Assumption 2	pseudo-SF to SF sk_y
3	$(\mathsf{sE}(x, \mathbf{w})s, \boxed{\text{random}})$	$\mathsf{rE}(0, y, \mathbf{w}; \mathbf{0})$		

Fig. 2. Sequence of games in the semi-functional space (the G_{p_2}-subgroup), where we drew a box to highlight the differences between each game and the preceding one, and games $2.i$.xx refer to the i'th secret key.

Auxiliary distributions. We consider the following auxiliary distributions for ciphertext and secret keys, where $(\mathsf{pp}, \mathsf{mpk}, \mathsf{msk}, \alpha, \mathbf{w})$ are sampled as in Setup.

- semi-functional (SF) master secret key: $\widehat{\mathsf{msk}} = (\boxed{g_1^{\alpha}}, \mathbf{w})$.
- normal ciphertexts:
$$\widehat{\mathsf{Enc}}(\mathsf{pp}, x; \mathsf{msk}, C), \quad C \leftarrow_{\mathrm{R}} \boxed{G_{p_1}}$$
this is identically distributed to real ciphertexts as computed using $\mathsf{Enc}(\mathsf{mpk}, x)$.
- semi-functional (SF) ciphertexts:
$$\widehat{\mathsf{Enc}}(\mathsf{pp}, x; \mathsf{msk}, \widehat{C}), \quad \widehat{C} \leftarrow_{\mathrm{R}} \boxed{G_{p_1} G_{p_2}}$$
- normal secret keys:
$$\widehat{\mathsf{KeyGen}}(\boxed{\mathsf{msk}}, y; R), \quad R \leftarrow_{\mathrm{R}} \boxed{G_{p_1}^* G_{p_3}}$$
this is identically distributed to real secret keys as computed using $\mathsf{KeyGen}(\mathsf{msk}, y)$.
- pseudo-normal secret keys:
$$\widehat{\mathsf{KeyGen}}(\boxed{\mathsf{msk}}, y; R), \quad R \leftarrow_{\mathrm{R}} \boxed{G_{p_1}^* G_{p_2}^* G_{p_3}}$$
- pseudo-semi-functional (pseudo-SF) secret keys:
$$\widehat{\mathsf{KeyGen}}(\boxed{\widehat{\mathsf{msk}}}, y; R), \quad R \leftarrow_{\mathrm{R}} \boxed{G_{p_1}^* G_{p_2}^* G_{p_3}}$$
- semi-functional (SF) secret keys:
$$\widehat{\mathsf{KeyGen}}(\boxed{\widehat{\mathsf{msk}}}, y; R), \quad R \leftarrow_{\mathrm{R}} \boxed{G_{p_1}^* G_{p_3}}$$

Remark 4 (decryption capabilities). Observe that all types of secret keys can decrypt a normal ciphertext. In addition, only normal and pseudo-normal secret keys can decrypt a semi-functional ciphertext, whereas pseudo-SF and SF keys cannot. The latter is consistent with the fact that we exploit α-hiding and $\mathsf{P}(x, y) = 0$ when we switch from pseudo-normal to pseudo-SF keys, which is precisely why we lose decryption capabilities in the G_{p_2}-components.

Game sequence. We present a series of games. We write $\mathsf{Adv}_{\mathrm{xx}}$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\mathrm{xx}}$.

- Game_0: is the real security game (c.f. Section 2.2).
- Game_1: is the same as Game_0 except that the challenge ciphertext is semi-functional. We also modify the distribution of κ_0 accordingly.
- $\mathsf{Game}_{2,i}$ for $i = 1,\ldots,q$: is the same as Game_1, except the first $i-1$ keys are semi-functional, and the last $q-i$ keys are normal. There are 4 sub-games, where the i'th key transitions from normal in $\mathsf{Game}_{2.i.0}$, to pseudo-normal in $\mathsf{Game}_{2.i.1}$, to pseudo-SF in $\mathsf{Game}_{2.i.2}$, to SF in $\mathsf{Game}_{2.i.3}$.
- Game_3: is the same as $\mathsf{Game}_{2,q,3}$, except that $\kappa_0 \leftarrow_{\mathrm{R}} \{0,1\}^{\lambda}$.

In Game_3, the view of the adversary $\mathcal{A}$ is statistically independent of the challenge bit β. Hence, $\mathsf{Adv}_3 = 0$. We complete the proof by establishing the following sequence of lemmas.

Lemma 1 (normal to semi-functional ciphertexts). *There exists $\mathcal{A}_1$ whose running time is roughly that of $\mathcal{A}$ such that*

$$|\mathsf{Adv}_0 - \mathsf{Adv}_1| \le \mathsf{Adv}^{\mathrm{SD1}}_{\mathcal{G},\mathcal{A}_1}(\lambda)$$

Proof. We will rely on Assumption 1. On input $D = (\mathbb{G}, g_1, g_3, g_1^{\alpha} g_2^{\alpha})$ and $T \in \{T_0, T_1\}$ where $T_0 \leftarrow_{\mathrm{R}} G_{p_1}$, $T_1 \leftarrow_{\mathrm{R}} G_{p_1} G_{p_2}$, the adversary $\mathcal{A}_1$ simulates $\mathcal{A}$ as follows:

Setup. Sample $\mathsf{H}, \mathbf{w}$ as in Setup, set $\mathsf{msk} := (g_1^{\alpha} g_2^{\alpha}, \mathbf{w})$ and output

$$\mathsf{pp} := (\mathbb{G}, \mathsf{H}, g_1, g_3) \quad \text{and} \quad \mathsf{mpk} := (g_1^{\mathbf{w}}, e(g_1, g_1^{\alpha} g_2^{\alpha})).$$

Ciphertext. Compute $(\mathsf{ct}_x, \kappa_0) \leftarrow \widehat{\mathsf{Enc}}(\mathsf{pp}, x; \mathsf{msk}, T)$.

Key Queries. On input the j'th key query y_j, output

$$\mathsf{sk}_j \leftarrow \widehat{\mathsf{KeyGen}}(\mathsf{msk}, y; g_1)$$

Output. Output whatever $\mathcal{A}$ outputs.

Observe that when $T = T_0 \leftarrow_{\mathrm{R}} G_{p_1}$, the output is identical to that in Game 0, and when $T = T_1 \leftarrow_{\mathrm{R}} G_{p_1} G_{p_2}$, the output is identical to that in Game 1. □

Lemma 2 (normal to pseudo-normal secret keys). *There exists $\mathcal{A}_2$ whose running time is roughly that of $\mathcal{A}$ such that for all i $= 1, 2, \ldots, q$,*

$$|\mathsf{Adv}_{2.i.0} - \mathsf{Adv}_{2.i.1}| \le \mathsf{Adv}^{\mathrm{SD2}}_{\mathcal{G},\mathcal{A}_2}(\lambda)$$

Proof. We will rely on Assumption 2. On input $D = (\mathbb{G}, g_1, g_3, g_{\{1,2\}}, g_{\{2,3\}})$ and $T \in \{T_0, T_1\}$ where $T_0 \leftarrow_{\mathrm{R}} G^*_{p_1} G_{p_3}$, $T_1 \leftarrow_{\mathrm{R}} G^*_{p_1} G^*_{p_2} G_{p_3}$, the adversary $\mathcal{A}_2$ simulates $\mathcal{A}$ as follows:

Setup. Sample $\alpha, \mathsf{H}, \mathbf{w}$ as in Setup, set

$$\mathsf{msk} := (g_1^{\alpha} \cdot g_{\{2,3\}}, \mathbf{w}) \quad \text{and} \quad \widehat{\mathsf{msk}} := (g_1^{\alpha}, \mathbf{w})$$

and output

$$\mathsf{pp} := (\mathbb{G}, \mathsf{H}, g_1, g_3) \quad \text{and} \quad \mathsf{mpk} := (g_1^{\mathbf{w}}, e(g_1, g_1^{\alpha})).$$

Ciphertext. Compute $(\mathsf{ct}_x, \kappa_0) \leftarrow \widehat{\mathsf{Enc}}(\mathsf{pp}, x; \mathsf{msk}, g_{\{1,2\}})$.

Key Queries. On input the j'th key query y_j, output

$$\mathsf{sk}_{y_j} \leftarrow \begin{cases} \widehat{\mathsf{KeyGen}}(\widehat{\mathsf{msk}}, y_j; g_1) & \text{if } j < i \quad \text{(semi-functional key)} \\ \widehat{\mathsf{KeyGen}}(\mathsf{msk}, y_j; T) & \text{if } j = i \quad \text{(normal vs pseudo-normal key)} \\ \widehat{\mathsf{KeyGen}}(\mathsf{msk}, y_j; g_1) & \text{if } j > i \quad \text{(normal key)} \end{cases}$$

Output. Output whatever $\mathcal{A}$ outputs.

Observe that when $T = T_0 \leftarrow_{\mathrm{R}} G_{p_1}^* G_{p_3}$, the output is identical to that in Game $2.i.0$, and when $T = T_1 \leftarrow_{\mathrm{R}} G_{p_1}^* G_{p_2}^* G_{p_3}$, the output is identical to that in Game $2.i.1$. □

Lemma 3 (pseudo-normal to pseudo-SF secret keys). *For all* $i = 1, 2, \ldots, q$,

$$|\mathsf{Adv}_{2.i.1} - \mathsf{Adv}_{2.i.2}| = 0$$

Proof. Observe that the only difference between Game $2.i.1$ and Game $2.i.2$ lies in the distribution of sk_{y_i}, which we sample using $\mathsf{msk} = g_1^\alpha g_2^\alpha$ and $\widehat{\mathsf{msk}} = g_1^\alpha$ respectively. This means the only difference between Game $2.i.1$ and Game $2.i.2$ lies in the G_{p_2}-component of sk_{y_i}, which are given by

$$g_2^{\mathsf{rE}(\alpha, y_i, \mathbf{w}; \mathbf{r})} \quad \text{and} \quad g_2^{\mathsf{rE}(0, y_i, \mathbf{w}; \mathbf{r})} \qquad (*)$$

respectively, where $\mathbf{r} \leftarrow_{\mathrm{R}} \mathcal{R}$. By the Chinese Remainder Theorem, it suffices to focus on the G_{p_2}-components of challenge ciphertext and secret keys, which are independent of the corresponding G_{p_1}-components. Observe that for all $j \neq i$, the G_{p_2}-component of sk_{y_j} is given by:

$$\begin{cases} \mathsf{rE}(0, y_j, \mathbf{w}; \mathbf{0}) = \mathsf{rE}(0, y_j, \mathbf{0}; \mathbf{0}) & \text{if } j < i \quad \text{(semi-functional key)} \\ \mathsf{rE}(\alpha, y_j, \mathbf{w}; \mathbf{0}) = \mathsf{rE}(\alpha, y_j, \mathbf{0}; \mathbf{0}) & \text{if } j > i \quad \text{(normal key)} \end{cases}$$

where the equality above follows by $\mathbf{w}$-hiding. This means that only the challenge ciphertext and the sk_{y_i} leaks any information about $\mathbf{w} \pmod{p_2}$. It now follows from the α-privacy property (modulo p_2) and $\mathsf{P}(x, y_i) = 0$ that

$$\mathsf{rE}(\alpha, y_i, \mathbf{w}; \mathbf{r}) \pmod{p_2} \quad \text{and} \quad \mathsf{rE}(0, y_i, \mathbf{w}; \mathbf{r}) \pmod{p_2}$$

are identically distributed from the view-point of the adversary. (Here, we also use secrecy of $\mathbf{r} \pmod{p_2}$.) This holds even if the adversary chooses y_i adaptively after seeing the challenge ciphertext ct_x, or if the challenge x is chosen after the adversary sees sk_{y_i} (c.f. Remark 1). □

Lemma 4 (pseudo-SF to SF secret keys). *There exists* $\mathcal{A}_3$ *whose running time is roughly that of* $\mathcal{A}$ *such that for all* $i = 1, 2, \ldots, q$,

$$|\mathsf{Adv}_{2.i.2} - \mathsf{Adv}_{2.i.3}| \leq \mathsf{Adv}_{\mathcal{G}, \mathcal{A}_3}^{\mathrm{SD2}}(\lambda)$$

Proof. We will again rely on Assumption 2. The proof is completely analogous to Lemma 2, except $\mathcal{A}_3$ uses $\widehat{\mathsf{msk}}$ instead of msk to sample sk_{y_j}.That is, $\mathcal{A}_3$ outputs

$$\mathsf{sk}_{y_j} \leftarrow \begin{cases} \widehat{\mathsf{KeyGen}}(\widehat{\mathsf{msk}}, y_j; g_1) & \text{if } j < i \quad \text{(semi-functional key)} \\ \widehat{\mathsf{KeyGen}}(\widehat{\mathsf{msk}}, y_j; T) & \text{if } j = i \quad \text{(pseudo-SF vs SF key)} \\ \widehat{\mathsf{KeyGen}}(\mathsf{msk}, y_j; g_1) & \text{if } j > i \quad \text{(normal key)} \end{cases}$$

Observe that when $T = T_1 \leftarrow_{\mathrm{R}} G^*_{p_1} G^*_{p_2} G_{p_3}$, the output is identical to that in Game 2.i.2, and when $T = T_0 \leftarrow_{\mathrm{R}} G^*_{p_1} G_{p_3}$, the output is identical to that in Game 2.i.3. □

Lemma 5 (final transition).

$$|\mathsf{Adv}_{3.q.3} - \mathsf{Adv}_4| \le 2^{-\Omega(\lambda)}$$

Proof. In Game 3.q.3, all the secret keys are semi-functional, which means they leak no information whatsoever about $\alpha \pmod{p_2}$. Next, let us examine the (semi-functional) challenge ciphertext. Observe that the quantity (from which the symmetric key κ_0 is derived)

$$e(\widehat{C}, g_1^\alpha) \cdot e(\widehat{C}, g_2^\alpha)$$

has $\log p_2 = \Theta(\lambda)$ bits of min-entropy as long as $\widehat{C} \in G_{p_1} G^*_{p_2}$, which occurs with probability $1 - 1/p_2$. Then, by the left-over hash lemma, $\kappa_0 = \mathsf{H}(e(\widehat{C}, g_1^\alpha) \cdot e(\widehat{C}, g_2^\alpha))$ is $2^{-\Omega(\lambda)}$-close to the uniform distribution over $\{0,1\}^\lambda$. The claim follows readily. □

5 Instantiations of Predicate Encodings

We present N-bilinear predicate encodings for a large class of predicates that have been considered in the literature. For concreteness, think of N as the order of the composite-order bilinear group. Note that in the proof of α-privacy, whenever we compute some value $v \neq 0 \in \mathbb{Z}_N$, we will simply assume that $\gcd(v, N) = 1$; otherwise, we will be able to compute a non-trivial factor of N. Instantiated via our framework, we obtain the adaptively-secure composite-order (H)IBE, ABE and spatial encryption schemes in [33, 36, 14]. In addition, we obtain novel (to the best of our knowledge) and simple constructions of adaptively-secure NIPE and doubly spatial encryption.

5.1 Inner Product (IPE)

Predicate [31]. Here, $\mathcal{X} = \mathcal{Y} := \mathbb{Z}_N^d$ and

$$\mathsf{P}(\mathbf{x}, \mathbf{y}) = 1 \text{ iff } \langle \mathbf{x}, \mathbf{y} \rangle = 0$$

First encoding (short secret keys) [7].

- $\mathcal{W} := \mathbb{Z}_N \times \mathbb{Z}_N^d; \mathcal{R} := \mathbb{Z}_N^*$.
- $\mathsf{sE}(\mathbf{x}, (u_0, \mathbf{u})) := (u_0 \mathbf{x} + \mathbf{u}, 1)$
- $\mathsf{rE}(\alpha, \mathbf{y}, (u_0, \mathbf{u}); r) := (r, \alpha - r \langle \mathbf{u}, \mathbf{y} \rangle)$

Second encoding (short ciphertext) [11].

- $\mathcal{W} := \mathbb{Z}_N \times \mathbb{Z}_N^d; \mathcal{R} := \mathbb{Z}_N \times \mathbb{Z}_N^*$.
- $\mathsf{sE}(\mathbf{x}, (u_0, \mathbf{u})) := (1, u_0 + \langle \mathbf{x}, \mathbf{u} \rangle)$
- $\mathsf{rE}(\alpha, \mathbf{y}, (u_0, \mathbf{u}); (r', r)) := (r\mathbf{u} - r'\mathbf{y}, r, \alpha - u_0 r)$

5.2 Non-Zero Inner Product (NIPE)

Predicate [2]. Here, $\mathcal{X} = \mathcal{Y} := \mathbb{Z}_N^d$ and

$$\mathsf{P}(\mathbf{x}, \mathbf{y}) = 1 \text{ iff } \langle \mathbf{x}, \mathbf{y} \rangle \neq 0$$

The constructions exploit the following simple algebraic fact: given $\mathbf{x}, \mathbf{y}, u_0\mathbf{x} + \mathbf{u}, \langle \mathbf{y}, \mathbf{w} \rangle$,

- if $\langle \mathbf{x}, \mathbf{y} \rangle \neq 0$, then we can recover u_0.
- if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$, then u_0 is perfectly random.

First encoding (short ciphertext).

- $\mathcal{W} := \mathbb{Z}_N^d; \mathcal{R} := \mathbb{Z}_N^*$.
- $\mathsf{sE}(\mathbf{x}, \mathbf{w}) := (\langle \mathbf{w}, \mathbf{x} \rangle, 1)$
- $\mathsf{rE}(\alpha, \mathbf{y}, \mathbf{w}; r) := (r, \alpha\mathbf{y} - r\mathbf{w})$

Second encoding (short secret keys).

- $\mathcal{W} := \mathbb{Z}_N \times \mathbb{Z}_N^d; \mathcal{R} := \mathbb{Z}_N^*$.
- $\mathsf{sE}(\mathbf{x}, (u_0, \mathbf{u})) := (u_0\mathbf{x} + \mathbf{u}, 1)$
- $\mathsf{rE}(\alpha, \mathbf{y}, (u_0, \mathbf{u}); r) := (r, \alpha + u_0 r, r\langle \mathbf{u}, \mathbf{y} \rangle)$

5.3 Spatial Encryption

Predicate [9]. Here, $\mathcal{X} := \mathbb{Z}_N^d, \mathcal{Y} := \mathbb{Z}_N^{d \times \ell}$ and

$$\mathsf{P}(\mathbf{x}, \mathbf{Y}) = 1 \text{ iff } \mathbf{x} \in \operatorname{span}(\mathbf{Y})$$

Recall from [9] that spatial encryption generalizes HIBE.

Supporting delegation. Consider a predicate P that supports delegation, namely, there is a partial ordering $\leq$ on $\mathcal{Y}$ such that for all $x \in \mathcal{X}$, the predicate $\mathsf{P}(x, \cdot)$ is monotone, i.e.

$$(y \leq y') \wedge \mathsf{P}(x, y) = 1 \implies \mathsf{P}(x, y') = 1.$$

For instance, in HIBE, $y \leq y'$ iff y' is a prefix of y. A bilinear encoding $(\mathsf{sE}, \mathsf{rE})$ for such a predicate supports delegation if given y, y' such that $y \leq y'$, we can efficiently compute a linear map L such that for all $(\alpha, \mathbf{w}, \mathbf{r}) \in \mathcal{D} \times \mathcal{W} \times \mathcal{R}$, L maps $(\mathbf{w}, \mathsf{rE}(\alpha, y', \mathbf{w}; \mathbf{r}))$ to $\mathsf{rE}(\alpha, y, \mathbf{w}; \mathbf{r})$. Note that we can always rerandomize the output due to linearity of receiver encoding.

Encoding (short ciphertext) [9, 11, 33, 14].

- $\mathcal{W} = \mathbb{Z}_N \times \mathbb{Z}_N^d; \mathcal{R} = \mathbb{Z}_N^*$.
- $\mathsf{sE}(\mathbf{x}, (u_0, \mathbf{u})) := (u_0 + \mathbf{u}^\top \mathbf{x}, 1)$
- $\mathsf{rE}(\alpha, \mathbf{Y}, (u_0, \mathbf{u}); r) := (r\mathbf{u}^\top \mathbf{Y}, -r, \alpha + r u_0)$

α-privacy holds for all $r \in \mathbb{Z}_N^*$, and relies on the fact that if $\mathbf{x} \notin \mathrm{span}(\mathbf{Y})$, then $\mathbf{u}^\top\mathbf{x}$ is statistically independent of $\mathbf{u}^\top\mathbf{Y}$ for a random $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_N^d$.

5.4 Doubly Spatial Encryption

Predicate [27]. Here, $\mathcal{X} := \mathbb{Z}_N \times \mathbb{Z}_N^{d \times \ell}, \mathcal{Y} := \mathbb{Z}_N^{d\times\ell'}$ and

$$\mathsf{P}((\mathbf{x}_0, \mathbf{X}), \mathbf{Y}) = 1 \text{ iff } (\mathbf{x}_0 + \mathrm{span}(\mathbf{X})) \cap \mathrm{span}(\mathbf{Y}) \neq \emptyset$$

Encoding [27].

- $\mathcal{W} = \mathbb{Z}_N \times \mathbb{Z}_N^d; \mathcal{R} = \mathbb{Z}_N^*$.
- $\mathsf{sE}((\mathbf{x}_0, \mathbf{X}), (u_0, \mathbf{u})) := (u_0 + \mathbf{u}^\top \mathbf{x}_0, \mathbf{u}^\top\mathbf{X}, 1)$
- $\mathsf{rE}(\alpha, \mathbf{Y}, (u_0, \mathbf{u}); r) := (r\mathbf{u}^\top \mathbf{Y}, -r, \alpha + r u_0)$

α-privacy holds for all $r \in \mathbb{Z}_N^*$, and relies on the fact that if $(\mathbf{x}_0 + \mathrm{span}(\mathbf{X})) \cap \mathrm{span}(\mathbf{Y}) = \emptyset$ then $\mathbf{u}^\top\mathbf{x}_0$ is statistically independent of $\mathbf{u}^\top\mathbf{X}, \mathbf{u}^\top\mathbf{Y}$ for a random $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_N^d$.

5.5 Attribute-Based Encryption (ABE)

We define (monotone) access structures using the language of (monotone) span programs [30].

Definition 1 (access structure [4, 30]). *A* (monotone) access structure *for attribute universe* $[n]$ *is a pair* $(\mathbf{M}, \rho)$ *where* $\mathbf{M}$ *is a* $\ell \times \ell'$ *matrix over* $\mathbb{Z}_N$ *and* $\rho : [\ell] \to [n]$. *Given* $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$, *we say that*

$$\mathbf{x} \textit{ satisfies } (\mathbf{M}, \rho) \textit{ iff } \mathbf{1} \in \textit{span}\langle \mathbf{M}_\mathbf{x} \rangle,$$

Here, $\mathbf{1} := (1, 0, \ldots, 0) \in \mathbb{Z}^{\ell'}$ *is a row vector;* $\mathbf{M}_\mathbf{x}$ *denotes the collection of vectors* $\{\mathbf{M}_j : x_{\rho(j)} = 1\}$ *where* $\mathbf{M}_j$ *denotes the j'th row of* $\mathbf{M}$*; and span refers to linear span of collection of (row) vectors over* $\mathbb{Z}_N$.

That is, $\mathbf{x}$ satisfies $(\mathbf{M}, \rho)$ iff there exists constants $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_N$ such that

$$\sum_{j : x_{\rho(j)} = 1} \omega_j \mathbf{M}_j = \mathbf{1}.$$

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

KP-ABE Predicate [26, 41]. Here, $\mathcal{X} := \mathbb{Z}_N^\ell, \mathcal{Y} := \{(\mathbf{M}, \rho) : \mathbf{M} \in \mathbb{Z}_N^{\ell \times \ell'}, \rho : [\ell] \to [\ell] \text{ is a permutation}\}$ (that is, $\ell = n$) and

$$\mathsf{P}(\mathbf{x}, (\mathbf{M}, \rho)) = 1 \text{ iff } \mathbf{x} \text{ satisfies } (\mathbf{M}, \rho)$$

Encoding. Our encoding improves upon that in [36] by reducing the length of rE (and thus the secret key size) from 2ℓ to $\ell+1$ elements.

- $\mathcal{W} = \mathbb{Z}_N^{\ell}; \mathcal{R} = \mathbb{Z}_N^{\ell'-1} \times \mathbb{Z}_N^*$.
- $\mathsf{sE}(\mathbf{x},\mathbf{w}) := (x_1 w_1, \ldots, x_\ell w_\ell, 1)$
- $\mathsf{rE}(\alpha, (\mathbf{M},\rho), \mathbf{w}; (\mathbf{u}, r)) := (\alpha_1 - r\, w_{\rho(1)}, \ldots, \alpha_\ell - r\, w_{\rho(\ell)}, r)$ where $\alpha_i := \mathbf{M}_i \binom{\alpha}{\mathbf{u}}$ is the i'th share of α and $\binom{\alpha}{\mathbf{u}}$ denotes the column vector in $\mathbb{Z}_N^{\ell'}$ formed by concatenating $\alpha \in Z_N$ and $\mathbf{u} \in Z_N^{\ell'-1}$.

In the prior construction [36], rE is given by

$$(\alpha_1 - r_1 w_{\rho(1)}, \ldots, \alpha_\ell - r_\ell w_{\rho(\ell)}, r_1, \ldots, r_\ell).$$

Here, α-privacy holds for all $r \in \mathbb{Z}_N^*$, and relies crucially on the fact that ρ is injective.

Remark 5 (GPSW encoding [26]). It is instructive here to revisit the encoding used in the selective ABE in [26] where rE is given by

$$(\alpha_1 / w_{\rho(1)}, \ldots, \alpha_\ell / w_{\rho(\ell)}).$$

This implies α-privacy but only in a statistical sense (the encoding only hides non-zero shares). Moreover, it does not satisfy $\mathbf{w}$-hiding.

CP-ABE Predicate [26, 17]. As before with $\mathcal{X}$ and $\mathcal{Y}$ switched, so that

$$\mathsf{P}((\mathbf{M},\rho), \mathbf{y}) = 1 \text{ iff } \mathbf{y} \text{ satisfies } (\mathbf{M},\rho)$$

Encoding. In the following encoding, we allow sE to be randomized:

- $\mathcal{W} = \mathbb{Z}_N^{\ell} \times \mathbb{Z}_N; \mathcal{R} = \mathbb{Z}_N^*$.
- $\mathsf{sE}((\mathbf{M},\rho), (\mathbf{w}, v); \mathbf{u}) := (1, w_{\rho(1)} + v_1, \ldots, w_{\rho(\ell)} + v_\ell)$ where $v_i := \mathbf{M}_i \binom{v}{\mathbf{u}}$ is the i'th share of v.
- $\mathsf{rE}(\alpha, \mathbf{y}, (\mathbf{w}, v); r) := (\alpha + r\,v, r, \{w_j r\}_{j : y_j = 1})$

Randomized sender encodings. We may handle the extension to randomized sender encodings where sE takes additional randomness $\mathbf{u}$ as follows:

- the requirement for α-privacy holds over random coin tosses of sE;
- affine sending encoding says that we can compute $g^{\mathsf{sE}(x,\mathbf{w})}$ given $g^{\mathbf{w}}$, x and the coin tosses used in sE;
- we extend the definition of Enc and $\widehat{\mathsf{Enc}}$ to use randomized sE in a straight-forward manner;
- the proof remains largely unchanged except for accounting for sender randomness when invoking α-privacy in the proof of Lemma 3.

Acknowledgments. I would like to thank Jie Chen, Kai-Min Chung, Yuval Ishai, Allison Lewko and Vinod Vaikuntanathan for insightful discussions.

References

[1] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM J. Comput.*, 36(4):845–888, 2006.

[2] N. Attrapadung and B. Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *Public Key Cryptography*, pages 384–402, 2010.

[3] N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *Public Key Cryptography*, pages 90–108, 2011.

[4] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution.* Ph.D., Technion - Israel Institute of Technology, 1996.

[5] M. Bellare, B. Waters, and S. Yilek. Identity-based encryption secure against selective opening attack. In *TCC*, pages 235–252, 2011.

[6] M. Bellare, V. T. Hoang, and P. Rogaway. Adaptively secure garbling with applications to one-time programs and secure outsourcing. In *ASIACRYPT*, pages 134–153, 2012.

[7] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[8] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[9] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pages 455–470, 2008.

[10] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pages 535–554, 2007.

[11] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

[12] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC*, pages 325–341, 2005.

[13] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.

[14] C. Chen, Z. Zhang, and D. Feng. Fully secure doubly-spatial encryption under simple assumptions. In *ProvSec*, pages 253–263, 2012.

[15] J. Chen and H. Wee. Dual system groups and its applications — compact HIBE and more. Full version in preparation, 2013. Preliminary version in [**?**].

[16] J. Chen and H. Wee. Fully, (almost) tightly secure IBE from standard assumptions. IACR Cryptology ePrint Archive, Report 2013/803, 2013. Preliminary version in [**?**].

[17] L. Cheung and C. C. Newport. Provably secure ciphertext policy ABE. In *ACM Conference on Computer and Communications Security*, pages 456–465, 2007.

[18] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.

[19] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002. Also, Cryptology ePrint Archive, Report 2001/085.

[20] U. Feige, J. Kilian, and M. Naor. A minimal model for secure computation. In *STOC*, pages 554–563, 1994.

[21] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO (2)*, pages 479–499, 2013. Also, Cryptology ePrint Archive, Report 2013/128.

[22] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[23] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[24] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. *J. Comput. Syst. Sci.*, 60(3):592–629, 2000.

[25] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554, 2013. Also, Cryptology ePrint Archive, Report 2013/337.

[26] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[27] M. Hamburg. *Spatial Encryption.* Ph.D., Stanford University, 2011. Also, Cryptology ePrint Archive, Report 2011/389.

[28] Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications. In *ISTCS*, pages 174–184, 1997.

[29] Y. Ishai and E. Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *FOCS*, pages 294–304, 2000.

[30] M. Karchmer and A. Wigderson. On span programs. In *Structure in Complexity Theory Conference*, pages 102–111, 1993.

[31] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.

[32] A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012. Also Cryptology ePrint Archive, Report 2011/490.

[33] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, pages 455–479, 2010.

[34] A. B. Lewko and B. Waters. Decentralizing attribute-based encryption. In *EUROCRYPT*, pages 568–588, 2011.

[35] A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, pages 180–198, 2012.

[36] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pages 62–91, 2010.

[37] A. B. Lewko, Y. Rouselakis, and B. Waters. Achieving leakage resilience through dual system encryption. In *TCC*, pages 70–88, 2011. Cryptology ePrint Archive, Report 2010/438.

[38] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[39] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.

[40] T. Okamoto and K. Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In *EUROCRYPT*, pages 591–608, 2012. Also, Cryptology ePrint Archive, Report 2011/543.

[41] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[42] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[43] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[44] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

[45] A. C.-C. Yao. Theory and applications of trapdoor functions. In *FOCS*, pages 80–91, 1982.