Functional Encryption for Inner Product Predicates from Learning with Errors

Shweta Agrawal[1*], David Mandell Freeman[2**], and Vinod Vaikuntanathan[3***]

[1] University of California, Los Angeles, USA; `shweta@cs.ucla.edu`
[2] Stanford University, USA; `dfreeman@cs.stanford.edu`.
[3] University of Toronto, Canada; `vinodv@cs.toronto.edu`

Abstract. We propose a lattice-based functional encryption scheme for inner product predicates whose security follows from the difficulty of the *learning with errors* (LWE) problem. This construction allows us to achieve applications such as range and subset queries, polynomial evaluation, and CNF/DNF formulas on encrypted data. Our scheme supports inner products over small fields, in contrast to earlier works based on bilinear maps.

Our construction is the first functional encryption scheme based on lattice techniques that goes beyond basic identity-based encryption. The main technique in our scheme is a novel twist to the identity-based encryption scheme of Agrawal, Boneh and Boyen (Eurocrypt 2010). Our scheme is weakly attribute hiding in the standard model.

Keywords. Functional encryption, predicate encryption, lattices, learning with errors.

1 Introduction

Traditional public key encryption is "coarse," in the sense that any user in the system can decrypt only messages encrypted with that user's public key. In a line of research beginning with the work of Sahai and Waters [37], a number of researchers have asked how to make encryption more fine-grained. The result is the notion of *functional encryption* [16], in which secret keys allow users to learn functions of encrypted data. Two important examples of functional encryption are *attribute-based encryption* (ABE) [37, 27] and *predicate encryption* (PE) [17, 28]. In (key-policy) ABE and PE systems, each ciphertext c is associated with an attribute a and each secret key s is associated with a predicate f. A user holding the key s can decrypt c if and only if $f(a) = 1$. The difference between the two types of systems is in the amount of

information revealed: an ABE system reveals the attribute associated with each ciphertext, while a PE system keeps the attribute hidden. (Formal definitions of these properties appear in Section 2.)

This hiding requirement has made predicate encryption systems much more difficult to construct than attribute-based encryption systems: while there exist ABE schemes that allow *any* access formula over attributes [34, 44], the most expressive PE scheme is that of Katz, Sahai, and Waters [28], who construct a PE scheme for *inner product predicates*. In such a scheme, attributes a and predicates f are expressed as vectors $\vec{v}_a$ and $\vec{w}_f$ respectively, and we say $f(a) = 1$ if and only if $\langle \vec{v}_a, \vec{w}_f \rangle = 0$. Despite this apparently restrictive structure, inner product predicates can support conjunction, subset and range queries on encrypted data [17] as well as disjunctions, polynomial evaluation, and CNF and DNF formulas [28].

All known constructions of attribute-based encryption [37, 27, 10, 21, 34, 26, 44, 8, 29, 33, 9] and predicate encryption [14, 1, 39, 17, 28, 40, 38, 32, 11, 29] make use of bilinear groups, and the security of these schemes is based on many different, and often complex, assumptions. In particular, there is at present no known construction of predicate encryption for inner products based on a "standard" assumption in bilinear groups.[4] As an example of a "nonstandard" assumption used in previous constructions, Katz, Sahai, and Waters present an assumption [28, Assumption 1] where the challenge consists of ten elements chosen in a specified way from a group whose order is the product of three large primes p, q, r, and the problem is to determine whether one of these elements has an order-q component. While assumptions such as this one can often be shown to hold in a suitable generic group model (e.g., [28, Appendix A]), to obtain more confidence in security we would like to build ABE and PE schemes based on computational problems whose complexity is better understood.

Our contribution. In this work we construct a lattice-based predicate encryption scheme for inner product predicates whose security follows from the difficulty of the *learning with errors* (LWE) problem. The LWE problem, in turn, is at least as hard as approximating the standard lattice problems GapSVP and SIVP in the *worst case* [36, 35] and is also conjectured to be difficult even for quantum adversaries. Our construction is the first functional encryption scheme based on lattice techniques that goes beyond basic identity-based encryption (which can be viewed as predicate encryption that tests equality on strings). Our construction is capable of instantiating all of the applications of predicate encryption proposed by Boneh and Waters [17] and Katz, Sahai, and Waters [28].[5]While our construction does not satisfy the strong notion of privacy defined by Katz, Sahai, and Waters [28], it does satisfy the slightly weaker notion considered by Okamoto and Takashima [32, 33] and Lewko *et al.* [29].

1.1 Overview of the Construction

Our approach. Just as functional encryption in bilinear groups builds on the ideas and techniques introduced in constructions of identity-based encryption (IBE) in bilinear

---

[4] Okamoto and Takashima [33] claim a PE construction from the decision linear assumption, but their paper only indicates how this is achieved for ABE.

[5] A detailed discussion of these applications can be found in the full version of this paper [2, §5].

groups [15, 25, 12, 13, 42, 22], our construction builds on the ideas and techniques used to achieve identity-based encryption from the LWE assumption [24, 5, 20, 3, 4]. However, there is a key difference between lattice IBE constructions (without random oracles) and bilinear group constructions that makes this kind of generalization more difficult in the lattice setting. Namely, in the bilinear group IBE constructions the groups remain fixed, while the ciphertexts and keys are manipulated so that group elements "cancel out" when a ciphertext matches a key. In the lattice IBE constructions, each key and ciphertext is constructed using a *different* lattice, and decryption only works when the key lattice and ciphertext lattice match. This structure does not easily generalize to the functional encryption setting, where each key may match many ciphertexts and each ciphertext may match many keys.

We solve this "lattice matching" problem using a new algebraic technique that builds on the IBE scheme of Agrawal, Boneh, and Boyen [3]. In our construction, we generate keys using a lattice Λ_f that depends only on the predicate f, and we generate ciphertexts c using a lattice Λ_a that depends only on the attribute a. Given a ciphertext c generated in this way and predicate f, we apply a suitable linear transformation that moves c into the lattice Λ_f if and only if $f(a) = 1$. Once this transformation is applied, we can decrypt using a key associated with Λ_f.

The details of our scheme and security proof are in Section 4. To prove security, we use a simulation technique that draws on ideas introduced in [3]. In particular, we construct our simulation using a "punctured" trapdoor that allows the simulator to answer secret key queries whenever $f(a) = 0$. In the simulation, we can use an LWE challenge to construct a ciphertext that either decrypts correctly or decrypts to a random message. While this technique suffices to prove that the system hides the message contents ("payload hiding"), it only allows us to prove a weak form of anonymity ("attribute hiding"). Specifically, given a ciphertext c and a number of keys that *do not* decrypt c, the user cannot determine the attribute associated with c. In the strong form of attribute hiding, the user cannot determine the attribute associated with c even when given keys that *do* decrypt c. (Formal definitions of these concepts appear in Section 2.) The weakened form of attribute hiding we do achieve is nonetheless more than is required for ABE and should be sufficient for many applications of PE.

Key technical ideas. Our encryption scheme is at its core based on the LWE scheme of Gentry, Peikert, and Vaikuntanathan [24, §7], which is itself a "dual" of the original Regev LWE scheme [36, §5]. From a geometric perspective, the public key in the GPV scheme describes a lattice Λ used to construct ciphertexts, and the secret key is derived from the dual lattice $\Lambda^{\perp}$. Existing constructions of lattice-based IBE in the standard model [5, 20, 3, 4] use the GPV encryption scheme but replace the fixed lattice Λ with a lattice Λ_{id} that depends on the user's identity id. Decryption only works when the ciphertext lattice Λ_{id} and secret key lattice $\Lambda_{\mathsf{id}'}$ are duals of each other, and there are several methods of ensuring that this is the case if and only if $\mathsf{id} = \mathsf{id}'$.

In trying to adapt these constructions to the predicate encryption setting, we run into the problem that each ciphertext can be decrypted by many secret keys and each secret key can decrypt many ciphertexts. Thus we cannot require that key lattices match ciphertext lattices in the same way as above.

3

Before explaining our solution to this problem, let us recall the selectively secure IBE scheme of Agrawal, Boneh, and Boyen [3]. In the ABB IBE scheme, the encryption lattice is constructed as

$$\Lambda_{\mathsf{id}} = \Lambda_q(\mathbf{A}_0 \parallel \mathbf{A}_1 + H(\mathsf{id})\mathbf{B}),$$

where $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}$ are $n \times m$ matrices over $\mathbb{Z}_q$ and $H(\mathsf{id})$ is a "full-rank difference" hash function. One can generate secret keys for $\Lambda_{\mathsf{id}}^{\perp}$ using a short basis of $\Lambda_q^{\perp}(\mathbf{A}_0)$ and the basis extension technique of [5, 20]. In the selective security proof, the LWE challenge is embedded as the matrix $\mathbf{A}_0$, and the matrix $\mathbf{A}_1 + H(\mathsf{id})\mathbf{B}$ is equipped with a "punctured" trapdoor that allows the simulator to respond to secret key queries for all identities id not equal to the challenge identity id^*.

The algebraic structure of the ABB IBE scheme gives the tools we need to solve the "lattice matching" problem described above. Specifically, in our predicate encryption scheme we encode an attribute vector $\vec{w} = (w_1, \ldots, w_\ell) \in \mathbb{Z}_q^\ell$ as the $n \times \ell m$ matrix

$$\mathbf{B}_{\vec{w}} := (w_1\mathbf{B}\parallel \cdots \parallel w_\ell\mathbf{B}).$$

where $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ is a uniformly random matrix chosen by the encryptor. We generate the ciphertext as a GPV encryption relative to the matrix

$$\Lambda_{\vec{w}} := \Lambda_q(\mathbf{A}_0 \parallel \mathbf{A}_1 + w_1\mathbf{B} \parallel \cdots \parallel \mathbf{A}_\ell + w_\ell\mathbf{B})$$

where the $\mathbf{A}_i$ are all $n \times m$ matrices. We view the ciphertext component that is close to $\Lambda_{\vec{w}}$ as a tuple $(\mathbf{c}_0, \ldots, \mathbf{c}_\ell) \in (\mathbb{Z}_q^m)^{\ell+1}$.

Since the recipient of a ciphertext does not know *a priori* which lattice was used to encrypt (indeed, this is exactly the anonymity property of predicate encryption), we cannot expect the recipient to possess a secret key derived from the dual of the ciphertext lattice as in the IBE case. Instead, we derive the key for a predicate vector $\vec{v}$ from the dual of a certain lattice $\Lambda_{\vec{v}}$ and apply a linear transformation $T_{\vec{v}}$ that moves the ciphertext into $\Lambda_{\vec{v}}$ exactly when $\langle \vec{v}, \vec{w} \rangle = 0$. If this linear transformation is "short" (in the sense of not increasing the length of vectors too much), then a GPV secret key derived from $\Lambda_{\vec{v}}^{\perp}$ can decrypt the ciphertext $T_{\vec{v}}(c)$.

Concretely, this transformation works as follows. For a predicate vector $\vec{v} = (v_1, \ldots, v_\ell) \in \mathbb{Z}_q^\ell$, we define the linear transformation $T_{\vec{v}} : (\mathbb{Z}_q^m)^{\ell+1} \rightarrow \mathbb{Z}_q^{2m}$ by

$$T_{\vec{v}}(\mathbf{c}_0, \ldots, \mathbf{c}_\ell) = (\mathbf{c}_0, \textstyle\sum_{i=1}^\ell v_i\mathbf{c}_i).$$

Some algebraic manipulation (detailed in Section 4) shows that applying this transformation to a ciphertext encrypted using $\Lambda_{\vec{w}}$ is equivalent to computing a GPV ciphertext using the lattice

$$\Lambda_{\vec{v},\vec{w}} := \Lambda_q\Big(\mathbf{A}_0 \parallel \sum_{i=1}^\ell v_i\mathbf{A}_i + \langle \vec{v}, \vec{w} \rangle\mathbf{B}\Big),$$

Letting the secret key for $\vec{v}$ be the GPV secret key associated to $\Lambda_q^{\perp}(\mathbf{A}_0 \parallel \sum_{i=1}^\ell v_i\mathbf{A}_i)$ allows the holder of a key for predicate $\vec{v}$ to decrypt a ciphertext associated with attribute $\vec{w}$ exactly when $\langle \vec{v}, \vec{w} \rangle = 0$. In this aspect our construction is inspired by that of Katz,

4

Sahai, and Waters [28]: the matrix $\mathbf{B}$ corresponds to the "masking terms" in a KSW ciphertext that "cancel out" exactly when $\langle \vec{v}, \vec{w} \rangle = 0$.

The reader may have observed that in the above formulation, the requirement that the linear transformation $T_{\vec{v}}$ be "short" implies that we cannot use all vectors $\vec{v} \in \mathbb{Z}_q^\ell$ as predicates, but only ones whose entries have small absolute value (when viewed as integers in $(-q/2, q/2]$). In Section 4 we will see how to get around this obstacle, enabling our construction to use arbitrary vectors in $\mathbb{Z}_q^\ell$. We do this by using the binary decomposition of the vector $\vec{v}$, at the expense of expanding the ciphertext by a factor of $\lg q$.

2 Predicate Encryption

We use the definition of predicate encryption proposed by Katz, Sahai, and Waters [28], which is based on the definition of *searchable encryption* proposed by Boneh and Waters [17]. We will let n denote the security parameter throughout this paper.

Definition 2.1 ([28, Definition 2.1]). A (key-policy) *predicate encryption scheme* for the class of predicates $\mathcal{F}$ over the set of attributes Σ consists of four ppt algorithms Setup, KeyGen, Enc, Dec such that:

- Setup takes as input a security parameter n and outputs a set of public parameters PP and a master secret key MK.
- KeyGen takes as input the master secret key MK and a (description of a) predicate $f \in \mathcal{F}$. It outputs a key sk_f.
- Enc takes as input the public parameters PP, an attribute $I \in \Sigma$, and a message M in some associated message space $\mathcal{M}$. It returns a ciphertext C.
- Dec takes as input a secret key sk_f and a ciphertext C. It outputs either a message M or the distinguished symbol $\bot$.

For correctness, we require that for all n, all $(\mathsf{PP}, \mathsf{MK})$ generated by $\mathsf{Setup}(1^n)$, all $f \in \mathcal{F}$, any key $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{MK}, f)$, all $I \in \Sigma$, and any ciphertext $C \leftarrow \mathsf{Enc}(\mathsf{PP}, I, M)$:

- If $f(I) = 1$, then $\mathsf{Dec}(\mathsf{sk}_f, C) = M$.
- If $f(I) = 0$, then $\mathsf{Dec}(\mathsf{sk}_f, C) = \bot$ with all but negligible probability.

In a *ciphertext-policy* scheme keys are associated with attributes and ciphertexts are associated with predicates; the syntax is otherwise the same.

Our construction in Section 4 satisfies a different correctness condition: If $f(I) = 1$ and $C = \mathsf{Enc}(\mathsf{PP}, I, M)$, then $\mathsf{Dec}(\mathsf{sk}_f, C) = M$, but if $f(I) = 0$ then $\mathsf{Dec}(\mathsf{sk}_f, C)$ is computationally indistinguishable from a uniformly random element in the message space $\mathcal{M}$. However, if $\mathcal{M}$ is exponentially large then we can easily transform our system into one satisfying Definition 2.1 by restricting the message space to some subset $\mathcal{M}' \subset \mathcal{M}$ with $|\mathcal{M}'|/|\mathcal{M}| = \mathrm{negl}(n)$.

2.1 Security

There are several notions of security for predicate encryption schemes. The most basic is *payload hiding*, which guarantees that no efficient adversary can obtain any information

about the encrypted message, but allows information about attributes to be revealed. A stronger notion is *attribute hiding*, which guarantees in addition that no efficient adversary can obtain any information about the attribute associated with a ciphertext. Following Lewko *et al.* [29, Definition 17], we also define an intermediate notion, *weak attribute hiding*, which makes the same guarantee only in the case that the adversary cannot decrypt the ciphertext. Our definition of security is "selective," in the sense that the adversary must commit to its challenge attributes before seeing any secret keys.

Definition 2.2 ([28, Definition 2.2]). A predicate encryption scheme with respect to $\mathcal{F}$ and Σ is *attribute hiding* if for all probabilistic polynomial-time adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is negligible in the security parameter n:

1. $\mathcal{A}(1^n)$ outputs $I_0, I_1 \in \Sigma$.
2. $\mathsf{Setup}(1^n)$ is run to generate PP and MK, and the adversary is given PP.
3. $\mathcal{A}$ may adaptively request keys for any predicates $f_1, \ldots, f_\ell \in \mathcal{F}$ subject to the restriction that $f_i(I_0) = f_i(I_1)$ for all i. In response, $\mathcal{A}$ is given the corresponding keys $\mathsf{sk}_{f_i} \leftarrow \mathsf{KeyGen}(\mathsf{MK}, f_i)$.
4. $\mathcal{A}$ outputs two equal-length messages M_0, M_1. If there is an i for which $f_i(I_0) = f_i(I_1) = 1$, then it is required that $M_0 = M_1$. A random bit b is chosen, and $\mathcal{A}$ is given the ciphertext $C \leftarrow \mathsf{Enc}(\mathsf{PP}, I_b, M_b)$.
5. The adversary may continue to request keys for additional predicates, subject to the same restrictions as before.
6. $\mathcal{A}$ outputs a bit b', and succeeds if $b' = b$. The *advantage* of $\mathcal{A}$ is the absolute value of the difference between its success probability and $1/2$.

We say the scheme is *weakly attribute hiding* if the same condition holds for adversaries $\mathcal{A}$ that are only allowed to request keys for predicates f_i with $f_i(I_0) = f_i(I_1) = 0$. We say the scheme is *payload hiding* if we require $I_0 = I_1$.

We observe that any scheme that is attribute hiding is weakly attribute hiding, and any scheme that is weakly attribute hiding is payload hiding. (In the payload hiding game no adversary can achieve non-negligible advantage when requesting a key for a predicate f with $f(I_0) = f(I_1) = 1$, so we may assume without loss of generality that the adversary does not request such a key.)

Remark 2.3. In our construction the spaces $\mathcal{F}$ of predicates and Σ of attributes depend on the public parameters PP output by Setup. We thus modify the security game so as to give the adversary descriptions of $\mathcal{F}$ and Σ before Step (1) and run the remainder of the game (including any remaining steps in the Setup algorithm) as described.

3 Lattice Preliminaries

In this section we collect the results from the literature that we will need for our construction and the proof of security. Results from probability that we need appear in the full version of this paper [2].

Notation. For any integer $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers modulo q and we represent $\mathbb{Z}_q$ as integers in $(-q/2, q/2]$. We let $\mathbb{Z}_q^{n \times m}$ denote the set of $n \times m$ matrices

with entries in $\mathbb{Z}_q$. We use bold capital letters (e.g. $\mathbf{A}$) to denote matrices, bold lowercase letters (e.g. $\mathbf{x}$) to denote vectors that are components of our encryption scheme, and arrows (e.g. $\vec{v}$) to denote vectors that represent attributes or predicates. The notation $\mathbf{A}^\intercal$ denotes the transpose of the matrix $\mathbf{A}$. When we say a matrix defined over $\mathbb{Z}_q$ has *full rank*, we mean that it has full rank modulo each prime factor of q.

3.1 Lattices

An *m-dimensional lattice* Λ is a full-rank discrete subgroup of $\mathbb{R}^m$. A *basis* of Λ is a linearly independent set of vectors whose span is Λ. We will usually be concerned with *integer lattices*, i.e., those whose points have coordinates in $\mathbb{Z}^m$. Among these lattices are the "q-ary" lattices defined as follows: for any integer $q \geq 2$ and any $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we define

$$\Lambda_q^\perp(\mathbf{A}) := \left\{ \mathbf{e} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{e} = \mathbf{0} \bmod q \right\}$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \left\{ \mathbf{e} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q \right\}.$$

The lattice $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is a coset of $\Lambda_q^\perp(\mathbf{A})$; namely, $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^\perp(\mathbf{A}) + \mathbf{t}$ for any $\mathbf{t}$ such that $\mathbf{A} \cdot \mathbf{t} = \mathbf{u} \bmod q$.

The Gram-Schmidt norm of a basis. Let $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_k\}$ be a set of vectors in $\mathbb{R}^m$. We use the following standard notation:

- $\|\mathbf{S}\|$ denotes the length of the longest vector in $\mathbf{S}$, i.e., $\max_{1 \leq i \leq k} \|\mathbf{s}_i\|$.
- $\widetilde{\mathbf{S}} := \{\tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_k\} \subset \mathbb{R}^m$ denotes the Gram-Schmidt orthogonalization of the vectors $\mathbf{s}_1, \ldots, \mathbf{s}_k$.

We refer to $\|\widetilde{\mathbf{S}}\|$ as the *Gram-Schmidt norm* of $\mathbf{S}$.

Ajtai [6] and later Alwen and Peikert [7] showed how to sample an almost uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ along with a basis $\mathbf{S}$ of $\Lambda_q^\perp((\mathbf{A}))$ with low Gram-Schmidt norm.

Theorem 3.1 ([7, Theorem 3.2] with $\delta = 1/3$). *Let q, n, m be positive integers with $q \geq 2$ and $m \geq 6n \lg q$. There is a probabilistic polynomial-time algorithm* $\mathsf{TrapGen}(q, n, m)$ *that with overwhelming probability (in n) outputs a pair $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{S} \in \mathbb{Z}^{m \times m})$ such that $\mathbf{A}$ is statistically close to uniform in $\mathbb{Z}_q^{n \times m}$ and $\mathbf{S}$ is a basis for $\Lambda_q^\perp((\mathbf{A}))$ satisfying*

$$\|\widetilde{\mathbf{S}}\| \leq O(\sqrt{n \log q}) \quad \text{and} \quad \|\mathbf{S}\| \leq O(n \log q).$$

Gaussian distributions. Let L be a discrete subset of $\mathbb{Z}^n$. For any vector $\mathbf{c} \in \mathbb{R}^n$ and any positive parameter $\sigma \in \mathbb{R}_{>0}$, let $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) := \exp\left(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / \sigma^2\right)$ be the Gaussian function on $\mathbb{R}^n$ with center $\mathbf{c}$ and parameter σ. Let $\rho_{\sigma,\mathbf{c}}(L) := \sum_{\mathbf{x} \in L} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$ be the discrete integral of $\rho_{\sigma,\mathbf{c}}$ over L, and let $\mathcal{D}_{L,\sigma,\mathbf{c}}$ be the discrete Gaussian distribution over L with center $\mathbf{c}$ and parameter σ. Specifically, for all $\mathbf{y} \in L$, we have $\mathcal{D}_{L,\sigma,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{y})}{\rho_{\sigma,\mathbf{c}}(L)}$. For notational convenience, $\rho_{\sigma,\mathbf{0}}$ and $\mathcal{D}_{L,\sigma,\mathbf{0}}$ are abbreviated as ρ_σ and $\mathcal{D}_{L,\sigma}$, respectively.

The following lemma gives a bound on the length of vectors sampled from a discrete Gaussian. The result follows from [31, Lemma 4.4], using [24, Lemma 5.3] to bound the smoothing parameter.

Lemma 3.2. *Let Λ be an n-dimensional lattice, let $\mathbf{T}$ be a basis for Λ, and suppose $\sigma \geq \|\widetilde{\mathbf{T}}\| \cdot \omega(\sqrt{\log n})$. Then for any $\mathbf{c} \in \mathbb{R}^n$ we have*

$$\Pr\left[\|\mathbf{x} - \mathbf{c}\| > \sigma\sqrt{n} : \mathbf{x} \xleftarrow{\text{R}} \mathcal{D}_{\Lambda,\sigma,\mathbf{c}}\right] \leq \text{negl}(n)$$

3.2 Sampling algorithms

We will also need the following algorithms to sample short vectors from specific lattices. Looking ahead, the algorithm SampleLeft [3, 20] will be used to sample keys in the real system, while the algorithm SampleRight [3] will be used to sample keys in the simulation.

Algorithm SampleLeft$(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, \mathbf{u}, \sigma)$**:**

> *Inputs:* a full rank matrix $\mathbf{A}$ in $\mathbb{Z}_q^{n \times m}$, a "short" basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$, a matrix $\mathbf{B}$ in $\mathbb{Z}_q^{n \times m_1}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter σ. $\qquad$ (3.1)
>
> *Output:* Let $\mathbf{F} := (\mathbf{A} \parallel \mathbf{B})$. The algorithm outputs a vector $\mathbf{e} \in \mathbb{Z}^{m+m_1}$ in the coset $\Lambda_q^{\mathbf{u}}(\mathbf{F})$.

Theorem 3.3 ([3, Theorem 17], [20, Lemma 3.2]). *Let $q > 2$, $m > n$ and $\sigma > \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m + m_1)})$. Then* SampleLeft$(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, \mathbf{u}, \sigma)$ *taking inputs as in (3.1), outputs a vector $\mathbf{e} \in \mathbb{Z}^{m+m_1}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}),\sigma}$ where $\mathbf{F} := (\mathbf{A} \parallel \mathbf{B})$.*

Algorithm SampleRight$(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T_B}, \mathbf{u}, \sigma)$**:**

> *Inputs:* matrices $\mathbf{A}$ in $\mathbb{Z}_q^{n \times k}$ and $\mathbf{R}$ in $\mathbb{Z}^{k \times m}$, a full rank matrix $\mathbf{B}$ in $\mathbb{Z}_q^{n \times m}$, a "short" basis $\mathbf{T_B}$ of $\Lambda_q^\perp(\mathbf{B})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter σ. $\qquad$ (3.2)
>
> *Output:* Let $\mathbf{F} := (\mathbf{A} \parallel \mathbf{A}\mathbf{R} + \mathbf{B})$. The algorithm outputs a vector $\mathbf{e} \in \mathbb{Z}^{2m}$ in the coset $\Lambda_q^{\mathbf{u}}(\mathbf{F})$.

Often the matrix $\mathbf{R}$ given to the algorithm as input will be a random matrix in $\{1, -1\}^{m \times m}$. Let S^m be the m-sphere $\{\mathbf{x} \in \mathbb{R}^{m+1} : \|\mathbf{x}\| = 1\}$. We define $s_R := \|\mathbf{R}\| = \sup_{\mathbf{x} \in S^{m-1}} \|\mathbf{R} \cdot \mathbf{x}\|$.

Theorem 3.4 ([3, Theorem 19]). *Let $q > 2, m > n$ and $\sigma > \|\widetilde{\mathbf{T_B}}\| \cdot s_R \cdot \omega(\sqrt{\log m})$. Then* SampleRight$(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T_B}, \mathbf{u}, \sigma)$ *taking inputs as in (3.2) outputs a vector $\mathbf{e} \in \mathbb{Z}^{2m}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}),\sigma}$ where $\mathbf{F} := (\mathbf{A} \parallel \mathbf{A}\mathbf{R} + \mathbf{B})$.*

3.3 The LWE Problem

The *Learning with Errors* problem, or LWE, is the problem of determining a secret vector over $\mathbb{F}_q$ given a polynomial number of "noisy" inner products. The decision variant is to distinguish such samples from random. More formally, we define the (average-case) problem as follows:

Definition 3.5 ([36]). Let $n \geq 1$ and $q \geq 2$ be integers, and let χ be a probability distribution on $\mathbb{Z}_q$. For $\mathbf{r} \in \mathbb{Z}_q^n$, let $A_{\mathbf{r},\chi}$ be the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{Z}_q$ according to χ, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{r} \rangle + e)$.

(a) The *search*-LWE$_{q,n,\chi}$ *problem* is: for uniformly random $\mathbf{r} \in \mathbb{Z}_q^n$, given a $\mathrm{poly}(n)$ number of samples from $A_{\mathbf{r},\chi}$, output $\mathbf{r}$.

(b) The *decision*-LWE$_{q,n,\chi}$ *problem* is: for uniformly random $\mathbf{r} \in \mathbb{Z}_q^n$, given a $\mathrm{poly}(n)$ number of samples that are either (all) from $A_{\mathbf{r},\chi}$ or (all) uniformly random in $\mathbb{Z}_q^n \times \mathbb{Z}_q$, output 0 if the former holds and 1 if the latter holds.

We say the decision-LWE$_{q,n,\chi}$ problem is *infeasible* if for all polynomial-time algorithms $\mathcal{A}$, the probability that $\mathcal{A}$ solves the decision-LWE problem (over $\mathbf{r}$ and $\mathcal{A}$'s random coins) is negligibly close to $1/2$ as a function of n.

The power of the LWE problem comes from the fact that for certain noise distributions χ, solving the search-LWE problem is as hard as finding approximate solutions to the shortest independent vectors problem (SIVP) and the decision version of the shortest vector problem (GapSVP) in the worst case. For polynomial size q there is a quantum reduction due to Regev, while for exponential size q there is a classical reduction due to Peikert. Furthermore, the search and decision versions of the problem are equivalent whenever q is a product of small primes. These results are summarized in the following:

Definition 3.6. For $\alpha \in (0,1)$ and an integer $q > 2$, let $\overline{\Psi}_\alpha$ denote the probability distribution over $\mathbb{Z}_q$ obtained by choosing $x \in \mathbb{R}$ according to the normal distribution with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$ and outputting $\lfloor qx \rceil$.

Theorem 3.7 ([36]). *Let n, q be integers and $\alpha \in (0,1)$ such that $q = \mathrm{poly}(n)$ and $\alpha q > 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that solves decision-*LWE$_{q,n,\overline{\Psi}_\alpha}$*, then there exists an efficient quantum algorithm that approximates* SIVP *and* GapSVP *to within $\tilde{O}(n/\alpha)$ in the worst case.*

Theorem 3.8 ([35]). *Let n, q be integers and $\alpha \in (0,1)$, and $q = \prod_i q_i \geq 2^{n/2}$, where the q_i are distinct primes satisfying $\omega(\sqrt{\log n})/\alpha \leq q_i \leq \mathrm{poly}(n)$. If there exists an efficient (classical) algorithm that solves decision-*LWE$_{q,n,\overline{\Psi}_\alpha}$*, then there exists an efficient (classical) algorithm that approximates* GapSVP *to within $\tilde{O}(n/\alpha)$ in the worst case.*

The following lemma will be used to show correctness of decryption.

Lemma 3.9 ([3, Lemma 12]). *Let $\mathbf{e}$ be some vector in $\mathbb{Z}^m$ and let $\mathbf{y} \leftarrow \overline{\Psi}_\alpha^m$. Then the quantity $|\langle \mathbf{e}, \mathbf{y} \rangle|$ when treated as an integer in $(-q/2, q/2]$ satisfies*

$$|\langle \mathbf{e}, \mathbf{y} \rangle| \leq \|\mathbf{e}\| q\alpha \cdot \omega(\sqrt{\log m}) + \|\mathbf{e}\|\sqrt{m}/2$$

with overwhelming probability (in m).

4 A Functional Encryption Scheme for Inner Product Predicates

In our system, each secret key will be associated with a predicate vector $\vec{v} \in \mathbb{Z}_q^\ell$ (for some fixed $\ell \geq 2$) and each ciphertext will be associated with an attribute vector $\vec{w} \in \mathbb{Z}_q^\ell$. Decryption should succeed if and only if $\langle \vec{v}, \vec{w} \rangle = 0 \pmod{q}$. Hence the predicate associated with the secret key is defined as $f_{\vec{v}}(\vec{w}) = 1$ if $\langle \vec{v}, \vec{w} \rangle = 0 \pmod{q}$, and $f_{\vec{v}}(\vec{w}) = 0$ otherwise.

4.1 The Construction

Let $n \in \mathbb{Z}^+$ be a security parameter and ℓ be the length of predicate and attribute vectors. Let $q = q(n, \ell)$ and $m = m(n, \ell)$ be positive integers. Let $\sigma = \sigma(n, \ell)$ and $\alpha = \alpha(n, \ell)$ be positive real Gaussian parameters. Define $k = k(n, \ell) := \lfloor \lg q \rfloor$. The encryption scheme described below encrypts a single bit; we describe how to encrypt multiple bits in the full version of this paper [2, §4.5].

LinFE.Setup$(1^n, 1^\ell)$**:** On input a security parameter n and a parameter ℓ denoting the length of predicate and attribute vectors, do:
1. Use the algorithm $\mathsf{TrapGen}(q, n, m)$ (from Theorem 3.1) to select a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a full-rank set of vectors $\mathbf{T_A} \subseteq \Lambda_q^\perp(\mathbf{A})$ such that $\|\widetilde{\mathbf{T_A}}\| \leq m \cdot \omega(\sqrt{\log m})$.
2. Choose $\ell \cdot (1 + k)$ uniformly random matrices $\mathbf{A}_{i,\gamma} \in \mathbb{Z}_q^{n \times m}$ for $i = 1, \ldots, \ell$ and $\gamma = 0, \ldots, k$.
3. Select a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$.
Output $\mathsf{PP} = (\mathbf{A}, \{\mathbf{A}_{i,\gamma}\}_{i \in \{1, \ldots, \ell\}, \gamma \in \{0, \ldots, k\}}, \mathbf{u})$ and $\mathsf{MK} = \mathbf{T_A}$.

LinFE.KeyGen$(\mathsf{PP}, \mathsf{MK}, \vec{v})$**:** On input the public parameters PP, the master secret key MK, and a predicate vector $\vec{v} = (v_1, \ldots, v_\ell) \in \mathbb{Z}_q^\ell$, do:
1. For $i = 1, \ldots, \ell$, let $\hat{v}_i$ be the integer in $[0, q-1]$ congruent to $v_i \bmod q$. Write the binary decomposition of $\hat{v}_i$ as

$$\hat{v}_i = \sum_{\gamma=0}^{k} v_{i,\gamma} \cdot 2^\gamma, \tag{4.1}$$

where $v_{i,\gamma}$ are in $\{0, 1\}$.
2. Define the matrices

$$\mathbf{C}_{\vec{v}} := \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{A}_{i,\gamma} \qquad \in \mathbb{Z}_q^{n \times m}$$

$$\mathbf{A}_{\vec{v}} := [\mathbf{A} \parallel \mathbf{C}_{\vec{v}}] \qquad \in \mathbb{Z}_q^{n \times 2m}.$$

3. Using the master secret key $\mathsf{MK} = (\mathbf{T_A}, \sigma)$, compute $\mathbf{e} \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{C}_{\vec{v}}, \mathbf{T_A}, \mathbf{u}, \sigma)$. Then $\mathbf{e}$ is a vector in $\mathbb{Z}^{2m}$ satisfying $\mathbf{A}_{\vec{v}} \cdot \mathbf{e} = \mathbf{u} \bmod q$.
Output the secret key $\mathsf{sk}_{\vec{v}} = \mathbf{e}$.

LinFE.Enc$(\mathsf{PP}, \vec{w}, M)$**:** On input public parameters PP, an attribute vector $\vec{w}$, and a message $M \in \{0, 1\}$, do:

    1. Choose a uniformly random matrix $\mathbf{B} \xleftarrow{\text{R}} \mathbb{Z}_q^{n \times m}$.

    2. Choose a uniformly random $\mathbf{s} \xleftarrow{\text{R}} \mathbb{Z}_q^n$.

    3. Choose a noise vector $\mathbf{x} \leftarrow \overline{\Psi}_\alpha^m$ and a noise term $x \leftarrow \overline{\Psi}_\alpha$.

    4. Compute $\mathbf{c}_0 \leftarrow \mathbf{A}^\intercal \mathbf{s} + \mathbf{x}$.

    5. For $i = 1, \ldots, \ell$ and $\gamma = 0, \ldots, k$, do the following:

        (a) Pick a random matrix $\mathbf{R}_{i,\gamma} \in \{-1, 1\}^{m \times m}$.

        (b) Compute $\mathbf{c}_{i,\gamma} \leftarrow (\mathbf{A}_{i,\gamma} + 2^\gamma w_i \mathbf{B})^\intercal \mathbf{s} + \mathbf{R}_{i,\gamma}^\intercal \mathbf{x}$.

    6. Compute $c' \leftarrow \mathbf{u}^\intercal \mathbf{s} + x + M \cdot \lfloor q/2 \rfloor \in \mathbb{Z}_q$.

    Output the ciphertext $\mathsf{CT} := (\mathbf{c}_0, \{\mathbf{c}_{i,\gamma}\}_{i \in \{1,\ldots,\ell\}, \gamma \in \{0,\ldots,k\}}, c')$.

LinFE.Dec$(\mathsf{PP}, \mathsf{sk}_{\vec{v}}, \mathsf{CT})$**:** On input the public parameters PP, a secret key $\mathsf{sk}_{\vec{v}}$ for predicate vector $\vec{v}$, and a ciphertext $\mathsf{CT} = (\mathbf{c}_0, \{\mathbf{c}_{i,\gamma}\}_{i \in \{1,\ldots,\ell\}, \gamma \in \{0,\ldots,k\}}, c')$, do:

    1. Define the binary expansion of the vector $\vec{v}$ as in (4.1), and compute

$$\mathbf{c}_{\vec{v}} := \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{c}_{i,\gamma}$$

    2. Let $\mathbf{c} := [\mathbf{c}_0 | \mathbf{c}_{\vec{v}}]$.

    3. Compute $z \leftarrow c' - \mathbf{e}^\intercal \mathbf{c} \pmod{q}$.

    Output 0 if $|z| < q/4$ (when interpreted as in integer in $(-q/2, q/2]$) and 1 otherwise.

For consistency with prior work, we choose the noise in Step **??** of Enc from the rounded continuous Gaussian $\overline{\Psi}_\alpha$. It was pointed out to us by a referee that one can instead use the discrete Gaussian $\mathcal{D}_{\mathbb{Z}, \alpha q}$ and obtain a system with the same security guarantee (up to a factor of $\sqrt{2}$); this result follows from [**?**, Lemma 2], using the work of Peikert [**?**].

4.2 Correctness

We now show that for certain parameter choices, if a bit M is encrypted to the attribute vector $\vec{w}$, the secret key $\mathbf{s}_{\vec{v}}$ corresponds to a predicate vector $\vec{v}$, and $\langle \vec{v}, \vec{w} \rangle = 0 \pmod{q}$, then the LinFE.Dec algorithm recovers M.

Lemma 4.1. *Suppose the parameters q and α are such that*

$$q/\lg q = \Omega\left(\sigma \cdot \ell \cdot m^{3/2}\right) \qquad and \qquad \alpha \leq \left(\log q \cdot \sigma \cdot \ell \cdot m \cdot \omega\sqrt{\log m}\right)^{-1}.$$

Let $\mathbf{e} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, \vec{v})$, $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{PP}, \vec{w}, M)$, and $\tilde{M} \leftarrow \mathsf{Dec}(\mathsf{PP}, \mathbf{e}, \mathsf{CT})$. If $\langle \vec{v}, \vec{w} \rangle = 0 \pmod{q}$, then with overwhelming probability we have $M' = M$.

Proof. During the first step of LinFE.Dec, we compute $\mathbf{c}_{\vec{v}}$, which is by definition:

$$\mathbf{c}_{\vec{v}} = \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{c}_{i,\gamma}.$$

This can be expanded as

$$
\mathbf{c}_{\vec{v}} = \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \left[(\mathbf{A}_{i,\gamma} + 2^{\gamma} w_i \mathbf{B})^{\mathsf{T}} \mathbf{s} + \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right]
$$

$$
= \left(\sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{A}_{i,\gamma} \right)^{\mathsf{T}} \mathbf{s} + \underbrace{\left(\sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} 2^{\gamma} v_{i,\gamma} w_i \right)}_{\vec{v} \cdot \vec{w}} \mathbf{B}^{\mathsf{T}} \mathbf{s} + \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x}.
$$

(4.2)

If $\langle \vec{v}, \vec{w} \rangle = 0 \pmod{q}$ then the middle term of (4.2) disappears, leaving

$$
\mathbf{c}_{\vec{v}} = \left(\sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{A}_{i,\gamma} \right)^{\mathsf{T}} \mathbf{s} + \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \pmod{q}
$$

In the second step of LinFE.Dec, we have:

$$
\mathbf{c} = [\mathbf{c}_0 | \mathbf{c}_{\vec{v}}] = \left[\mathbf{A} \,\bigg\|\, \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{A}_{i,\gamma} \right]^{\mathsf{T}} \mathbf{s} + \left[\mathbf{x} \,\bigg|\, \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \pmod{q}
$$

$$
= \mathbf{A}_{\vec{v}}^{\mathsf{T}} \cdot \mathbf{s} + \left[\mathbf{x} \,\bigg|\, \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \pmod{q}
$$

In the third step of LinFE.Dec, we multiply $\mathbf{c}$ with the key $\mathbf{e}$. Recall that by Theorem 3.3 we have $\mathbf{A}_{\vec{v}} \cdot \mathbf{e} = \mathbf{u} \pmod{q}$. It follows that

$$
\mathbf{e}^{\mathsf{T}} \mathbf{c} = \mathbf{u}^{\mathsf{T}} \mathbf{s} + \mathbf{e}^{\mathsf{T}} \left[\mathbf{x} \,\bigg|\, \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \pmod{q}.
$$

Finally, we compute:

$$
z = c' - \mathbf{e}^{\mathsf{T}} \mathbf{c} \pmod{q}
$$

$$
= (\mathbf{u}^{\mathsf{T}} \mathbf{s} + x + M \cdot \lfloor q/2 \rceil) - \mathbf{u}^{\mathsf{T}} \mathbf{s} - \mathbf{e}^{\mathsf{T}} \left[\mathbf{x} \,\bigg|\, \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \pmod{q}
$$

$$
= M \cdot \lfloor q/2 \rceil + \underbrace{\left(x - \mathbf{e}^{\mathsf{T}} \left[\mathbf{x} \,\bigg|\, \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \right)}_{\text{low-norm noise}} \pmod{q}
$$

To obtain $\tilde{M} = M$, it suffices to set the parameters so that with overwhelming probability,

$$
\left| x - \mathbf{e}^{\mathsf{T}} \left[\mathbf{x} \,\bigg|\, \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \right| < q/4.
$$

(4.3)

Writing $\mathbf{e} = [\mathbf{e}_1 | \mathbf{e}_2]$ with $\mathbf{e}_i \in \mathbb{Z}^m$ allows us to rewrite this "noise" term as

$$x - \left(\mathbf{e}_1 + \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma} \mathbf{e}_2 \right)^{\top} \mathbf{x}.$$

By Theorem 3.3 and Lemma 3.2, we have $\|\mathbf{e}\| < \sigma\sqrt{2m}$ with overwhelming probability. By [3, Lemma 15], we have $\|\mathbf{R}_{i,\gamma} \cdot \mathbf{e}_2\| \leq 12\sqrt{2m} \cdot \|\mathbf{e}_2\|$ with overwhelming probability. Since $v_{i,\gamma} \in \{0, 1\}$ it follows that

$$\left\| \mathbf{e}_1 + \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma} \mathbf{e}_2 \right\| < \left(1 + 12 \cdot \ell \cdot (1 + k) \cdot \sqrt{2m} \right) \cdot \sigma\sqrt{2m} = O(\ell \cdot k \cdot \sigma \cdot m).$$

It now follows from Lemma 3.9 that the error term (4.2) has absolute value at most

$$\left(q\alpha \cdot \omega(\sqrt{\log m}) + \sqrt{m}/2 \right) \cdot O\left(\ell \cdot \sigma \cdot m \cdot \lg q \right). \tag{4.4}$$

(Recall that $k = \lfloor \lg q \rfloor$.) For the quantity (4.4) to have absolute value less than $q/4$, it suffices to choose q and α as in the statement of the Lemma. $\qquad\square$

4.3 Security

We use the simulation technique of Agrawal, Boneh, and Boyen [3] to reduce the security of our system to the hardness of the decision-LWE problem.

Theorem 4.2. *Suppose $m \geq 6n \log q$. If the decision-$\mathsf{LWE}_{q,\alpha}$ problem is infeasible, then the predicate encryption scheme described above is weakly attribute hiding.*

To prove the theorem we define a series of three games against an adversary $\mathcal{A}$ that plays the weak attribute hiding game (subject to the modification described in Remark 2.3). The adversary $\mathcal{A}$ outputs two attribute vectors $\vec{w}_0$ and $\vec{w}_1$ at the beginning of each game, and at some point outputs two messages M_0 and M_1. Each game comes in two variants, reflecting the choice of attribute/message pair used to create the challenge ciphertext. The first game correspond to the real security game. In the intermediate games we use "alternative" setup, key generation, and encryption algorithms $\mathsf{Sim.Setup}$, $\mathsf{Sim.KeyGen}$, and $\mathsf{Sim.Enc}$. The algorithm $\mathsf{Sim.Setup}$ takes as additional input an attribute vector $\vec{w}^*$, and $\mathsf{Sim.Enc}$ takes as additional input the master key output by $\mathsf{Sim.Setup}$. Recall that during the course of the game the adversary can only request keys for predicate vectors $\vec{v}$ such that $\langle \vec{v}, \vec{w}_0 \rangle \neq 0$ and $\langle \vec{v}, \vec{w}_1 \rangle \neq 0$.

$\mathsf{Game}_{0,b}$: For $b \in \{0, 1\}$, the challenger runs the $\mathsf{LinFE.Setup}$ algorithm, answers the adversary's secret key queries using the $\mathsf{LinFE.KeyGen}$ algorithm, and generates the challenge ciphertext using the $\mathsf{LinFE.Enc}$ algorithm with attribute $\vec{w}_b$ and message M_b.

$\mathsf{Game}_{1,b}$: For $b \in \{0, 1\}$, the challenger runs the $\mathsf{Sim.Setup}$ algorithm with $\vec{w}^* = \vec{w}_b$ and answers the adversary's secret key queries using the $\mathsf{Sim.KeyGen}$ algorithm. The challenger generates the challenge ciphertext using the $\mathsf{Sim.Enc}$ algorithm with attribute $\vec{w}_b$ and message M_b.

$\mathsf{Game}_{2,b}$: This game is the same as $\mathsf{Game}_{1,b}$, except the challenger generates the challenge ciphertext by choosing a uniformly random element of the ciphertext space.

We now define the alternative setup, key generation, and encryption algorithms.

Sim.Setup$(1^n, 1^\ell, \vec{w}^*)$**:** On input a security parameter n, a parameter ℓ denoting the length of predicate and attribute vectors, and an attribute vector $\vec{w}^* \in \mathbb{Z}_q^\ell$, do the following:

1. Choose a random matrix $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$ and a random vector $\mathbf{u} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^n$.
2. Use $\mathsf{TrapGen}(q, n, m)$ to generate a matrix $\mathbf{B}^* \in \mathbb{Z}_q^{n \times m}$ along with a basis $\mathbf{T}_{\mathbf{B}^*} \in \mathbb{Z}^{m \times m}$ of $\Lambda_q^\perp(\mathbf{B}^*)$.
3. For $i = 1, \ldots, \ell$ and $\gamma = 0, \ldots, k$, pick random matrices $\mathbf{R}_{i,\gamma}^* \xleftarrow{\mathrm{R}} \{-1, 1\}^{m \times m}$ and set
$$\mathbf{A}_{i,\gamma} \leftarrow \mathbf{A}\,\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^*\,\mathbf{B}^*.$$

Output the public parameters and master key

$$\mathsf{PP} = \big(\mathbf{A}, \{\mathbf{A}_{i,\gamma}\}_{i \in \{1,\ldots,\ell\}, \gamma \in \{0,\ldots,k\}}, \mathbf{u}\big), \quad \mathsf{MK} = \big(\vec{w}^*, \{\mathbf{R}_{i,\gamma}^*\}_{i \in \{1,\ldots,\ell\}, \gamma \in \{0,\ldots,k\}}, \mathbf{B}^*, \mathbf{T}_{\mathbf{B}^*}\big)$$

Sim.KeyGen$(\mathsf{PP}, \mathsf{MK}, \vec{v})$**:** On input a master key MK and a vector $\vec{v} \in \mathbb{Z}_q^\ell$, do the following:

1. If $\langle \vec{v}, \vec{w}^* \rangle = 0$, output $\perp$.
2. Define the r-ary decomposition of v_i as in (4.1).
3. Define the matrices
$$\mathbf{C}_{\vec{v}} := \sum_{i=1}^\ell \sum_{\gamma=0}^k v_{i,\gamma} \mathbf{A}_{i,\gamma} \in \mathbb{Z}_q^{n \times m}, \qquad \mathbf{A}_{\vec{v}} := [\mathbf{A} \parallel \mathbf{C}_{\vec{v}}] \in \mathbb{Z}_q^{n \times 2m}.$$

Observe that
$$\mathbf{A}_{\vec{v}} = \left[\mathbf{A} \,\middle\|\, \mathbf{A}\Big(\sum_{i=1}^\ell \sum_{\gamma=0}^k v_{i,\gamma} \mathbf{R}_{i,\gamma}^*\Big) - \underbrace{\Big(\sum_{i=1}^\ell \sum_{\gamma=0}^k 2^\gamma v_{i,\gamma} w_i^*\Big)}_{\langle \vec{v}, \vec{w}^* \rangle} \mathbf{B}^*\right].$$

4. Let $\mathbf{e} \leftarrow \mathsf{SampleRight}\Big(\mathbf{A},\ -\langle \vec{v}, \vec{w}^* \rangle \mathbf{B}^*,\ \sum_{i=1}^\ell \sum_{\gamma=0}^k v_{i,\gamma} \mathbf{R}_{i,\gamma}^*,\ \mathbf{T}_{\mathbf{B}^*},\ \mathbf{u},\ \sigma\Big) \in \mathbb{Z}_q^{2m}$.

Output the secret key $\mathsf{sk}_{\vec{v}} = \mathbf{e}$.

Sim.Enc$(\mathsf{PP}, \vec{w}, M, \mathsf{MK})$**:** This algorithm is the same as the $\mathsf{LinFE.Enc}$ algorithm, except:

1. In Step 1, matrix $\mathbf{B}^* \in \mathsf{MK}$ is used instead of a random matrix $\mathbf{B}$.

14

2. In Step 5a, the matrices $\mathbf{R}^*_{i,\gamma} \in \mathsf{MK}$ for are used instead of random matrices $\mathbf{R}_{i,\gamma}$ for $i = 1, \ldots, \ell$ and $\gamma = 0, \ldots, k$.

To prove security of our system, we show that each pair of games $(\mathsf{Game}_i, \mathsf{Game}_{i+1})$ are either statistically indistinguishable or computationally indistinguishable under the decision-LWE assumption.

Lemma 4.3. *For a given $b \in \{0,1\}$, the view of the adversary $\mathcal{A}$ in $\mathsf{Game}_{0,b}$ is statistically close to the view of $\mathcal{A}$ in $\mathsf{Game}_{1,b}$.*

The proof of Lemma 4.3 can be found in the full version of this paper [2].

Lemma 4.4. *For a given $b \in \{0,1\}$, if the decision-LWE assumption holds, then the view of the adversary $\mathcal{A}$ in $\mathsf{Game}_{1,b}$ is computationally indistinguishable from the view of $\mathcal{A}$ in $\mathsf{Game}_{2,b}$.*

Proof. Suppose we are given $m + 1$ LWE challenges $(\mathbf{a}_i, y_i) \in \mathbb{Z}^n_q \times \mathbb{Z}_q$ for $j = 0, \ldots, m$, where either $y_j = \langle \mathbf{a}_j, \mathbf{s} \rangle + x_j$ for some (fixed) random secret $\mathbf{s} \xleftarrow{\text{R}} \mathbb{Z}^n_q$ and discrete Gaussian noise $x_j \leftarrow \overline{\Psi}_\alpha$, or y_j is uniformly random in $\mathbb{Z}_q$ (and this choice is the same for each challenge). We define the following variables:

$$
\mathbf{A} := \begin{pmatrix} | & & | \\ \mathbf{a}_1 & \cdots & \mathbf{a}_m \\ | & & | \end{pmatrix} \in \mathbb{Z}^{n \times m}_q \quad \mathbf{u} := \mathbf{a}_0 \tag{4.5}
$$
$$
\mathbf{c}_0 := (y_1, \ldots, y_m) \in \mathbb{Z}^m_q \qquad c' := y_0 + M_b \cdot \lfloor \tfrac{q}{2} \rceil
$$

We simulate the challenger as follows:

- **Setup:** Run Sim.Setup with $\vec{w}^* = \vec{w}_b$, and let $\mathbf{A}$ and $\mathbf{u}$ be as in (4.5).
- **Private key queries:** Run the Sim.KeyGen algorithm.
- **Challenge ciphertext:** For $i = 1, \ldots, \ell$ and $\gamma = 0, \ldots, k$, let $\mathbf{c}_{i,\gamma} = \mathbf{R}^*_{i,\gamma}{}^\mathsf{T}\mathbf{c}_0$ (using $\mathbf{R}^*_{i,\gamma} \in \mathsf{MK}$). Output $(\mathbf{c}_0, \{\mathbf{c}_{i,\gamma}\}_{i \in \{1,\ldots,\ell\}, \gamma \in \{0,\ldots,k\}}, c')$.

  Now observe that for $i = 1, \ldots, \ell$ and $\gamma = 0, \ldots, k$, the Sim.Enc algorithm sets

  $$
  \mathbf{c}_{i,\gamma} = \left(\mathbf{A}\mathbf{R}_{i,\gamma} - 2^\gamma w^*_i \mathbf{B}^* + 2^\gamma w^*_i \mathbf{B}^* \right)^\mathsf{T} \mathbf{s} + \mathbf{R}^*_{i,\gamma}{}^\mathsf{T}\mathbf{x} = \mathbf{R}^*_{i,\gamma}{}^\mathsf{T}(\mathbf{A}^\mathsf{T}\mathbf{s} + \mathbf{x}).
  $$

It follows that if $y_j = \langle \mathbf{a}_j, \mathbf{s} \rangle + x_j$, then $\mathbf{c}_{i,\gamma} = \mathbf{R}^*_{i,\gamma}{}^\mathsf{T}\mathbf{c}_0$ and the simulator described above is identical to the challenger in $\mathsf{Game}_{1,b}$.

On the other hand, if y_j is random in $\mathbb{Z}_q$, then the simulated ciphertext is $(\mathbf{c}_0, \overline{\mathbf{R}}^*{}^\mathsf{T}\mathbf{c}_0, c')$, where $\overline{\mathbf{R}}^*$ is the concatenation of the matrices $\mathbf{R}^*_{i,\gamma}$. By the standard leftover hash lemma (e.g. [41, Theorem 8.37]), the quantities $\mathbf{A}\overline{\mathbf{R}}^*$ and $\overline{\mathbf{R}}^*{}^\mathsf{T}\mathbf{c}_0$ are independent uniformly random samples. Thus in this case the ciphertext is uniformly random and the simulator described above is identical the challenger in $\mathsf{Game}_{2,b}$.

We conclude that any efficient adversary that can distinguish $\mathsf{Game}_{1,b}$ from $\mathsf{Game}_{2,b}$ can solve the decision-LWE problem. $\square$

15

Lemma 4.5. *The view of the adversary $\mathcal{A}$ in* $\mathsf{Game}_{2,0}$ *is statistically indistinguishable from the view of $\mathcal{A}$ in* $\mathsf{Game}_{2,1}$.

Proof. Note that the only place where $\vec{w}^*$ appears in $\mathsf{Game}_{2,b}$ is in the public parameter $\mathbf{A}_{i,\gamma} := \mathbf{A}\mathbf{R}^*_{i,\gamma} - 2^\gamma w^*_i \mathbf{B}^*$. Let $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n \times m\ell(k+1)}$ and $\overline{\mathbf{R}}^* \in \mathbb{Z}_q^{m \times m\ell(k+1)}$ be the concatenation of the $\mathbf{A}_{i,\gamma}$ and the $\mathbf{R}^*_{i,\gamma}$, respectively. Then we have $\overline{\mathbf{A}} = \mathbf{A}\overline{\mathbf{R}}^*$. By Lemma [3, Lemma 13] the pair $(\mathbf{A}, \mathbf{A}\overline{\mathbf{R}}^*)$ is statistically indistinguishable from $(\mathbf{A}, \mathbf{C})$ where $\mathbf{C}$ is uniformly random. Since for any fixed value of $\mathbf{X}$ and uniformly random $\mathbf{C}$, the variable $\mathbf{C} - \mathbf{X}$ is also uniformly random, it follows that the distributions of $\mathbf{A}_{i,\gamma}$ in the two games are statistically indistinguishable.

Theorem 4.2 now follows from a simple hybrid argument using the results of Lemmas 4.3, 4.4, and 4.5; details are in the full version of this paper [2].

4.4 Parameter Selection

We can extract from the above description the parameters required for correctness and security of the system. For correctness of decryption, by Lemma 4.1 we require

$$q/\lg q = \Omega\left(\sigma \cdot \ell \cdot m^{3/2}\right) \qquad \text{and} \qquad \alpha \le \left(\log q \cdot \sigma \cdot \ell \cdot m \cdot \omega\sqrt{\log m}\right)^{-1}. \quad (4.6)$$

In our security theorem (Theorem 4.2), we require $m > 6n \lg q$ in order for the output of $\mathsf{TrapGen}$ to be statistically random. The additional constraints imposed by our security reduction are the following:

- From the description of $\mathsf{LinFE.Setup}$ and $\mathsf{LinFE.KeyGen}$, we have $\|\widetilde{\mathbf{T_A}}\| = O(\sqrt{n\log q})$ (by Theorem 3.1) and $\mathbf{e} \leftarrow D_{\Lambda_q^{\mathsf{u}}(\mathbf{A}||\mathbf{B}),\sigma}$ (by Theorem 3.3), subject to the requirement that

$$\sigma \ge \|\widetilde{\mathbf{T}}_{\mathbf{A}}\| \cdot \omega(\sqrt{\log m}) = O(\sqrt{n\log q}) \cdot \omega(\sqrt{\log m}).$$

- From the description of $\mathsf{Sim.Setup}$ and $\mathsf{Sim.KeyGen}$, we have $\|\widetilde{\mathbf{T}_{\mathbf{B}^*}}\| = O(\sqrt{n\log q})$ (by Theorem 3.1), and $\mathbf{e} \leftarrow D_{\Lambda_q^{\mathsf{u}}(\mathbf{A}||\mathbf{B}),\sigma}$ (by Theorem 3.4), subject to the requirement that

$$\sigma \ge \|\widetilde{\mathbf{T}}_{\mathbf{B}^*}\| \cdot s_R \cdot \omega(\sqrt{\log m}) \qquad (4.7)$$

Since $\mathbf{R}$ is a sum of $\ell \cdot (\lg q + 1)$ random matrices with $\{1, -1\}$ entries, it follows from [3, Lemma 15] that $s_R = \sup_{\{\mathbf{x}: \|\mathbf{x}\|=1\}} \|\mathbf{R}\mathbf{x}\| = O(\ell \cdot (\lg q + 1) \cdot \sqrt{m})$ with overwhelming probability. Plugging this value into (4.7), we see that it suffices to choose

$$\sigma \ge O(\sqrt{n\log q}) \cdot O(\ell \cdot (\lg q + 1) \cdot \sqrt{m}) \cdot \omega(\sqrt{\log m}).$$

Thus, to satisfy the more stringent of the above two conditions (i.e., the latter), we set

$$\sigma = \omega(m \cdot \ell \cdot \log q \cdot \sqrt{\log m}), \qquad (4.8)$$

16

using the fact (noted above) that $m \geq 6n \log q$.

In order to reduce decision-LWE to approximating worst-case lattice problems to within $\text{poly}(n)$ factors we have two options: for polynomial size q we can use Regev's quantum reduction (Theorem 3.7) with $q\alpha > 2\sqrt{n}$ and $\alpha \geq 1/\text{poly}(n)$, while for exponential size q we can use Peikert's classical reduction (Theorem 3.8) with each prime factor q_i of q satisfying $\omega(\sqrt{\log n})/\alpha < q_i < \text{poly}(n)$. (Note that a large value of q may be required for certain applications; see the full version of this paper [2, §5] for details.)

The following selection of parameters satisfies all of these constraints. For a given ℓ, pick a small constant $\delta > 0$, and set

$$
\begin{array}{lll}
m = \lceil n^{1+\delta} \rceil, & & \text{to satisfy } m > 6n \lg q \\
\sigma = \lceil n^{2+2\delta} \cdot \ell \rceil, & & \text{to satisfy (4.8)} \\
q_i = \text{the } i\text{th prime larger than } (\ell \log \ell)^2 \cdot n^{7/2+5\delta} & & \\
\alpha = \Omega \left((\ell \log \ell)^2 \cdot n^{3+5\delta} \right)^{-1} & & \text{to satisfy (4.6)}
\end{array}
$$

Observe that the above setting of parameters satisfies the conditions for applying Theorems 3.7 and 3.8. To obtain polynomial size q we use $q = q_1$, while to obtain exponential size q we use $q = \prod_{i=1}^{\tau} q_i$, where τ is chosen so that $q > 2^{n/2}$. In either case we can choose δ large enough so that $n^{1+\delta} > 6n \lg q$. In the former case, the security of the scheme can be based on the hardness of approximating SIVP and GapSVP to within a factor of $\tilde{O}(n/\alpha) = \tilde{O}((\ell \log \ell)^2 \cdot n^{4+5\delta})$ in the worst case (by quantum algorithms). In the latter case, security is based on the hardness of approximating GapSVP to within a factor of $\tilde{O}(n/\alpha) = \tilde{O}((\ell \log \ell)^2 \cdot n^{4+5\delta})$ in the worst case (by classical algorithms).

Note that since $m > n \lg q$ and $q_i > n$, the matrices $\mathbf{A}$ and $\mathbf{B}$ have full rank modulo each prime divisor of q with overwhelming probability, as required for successful execution of the SampleLeft and SampleRight algorithms.

Finally, we note that these parameter choices are not necessarily optimal, and one might be able to set the parameters to have somewhat smaller values while maintaining correctness and security. In particular, one might be able to reduce the ciphertext size by using the r-ary expansion of the vector $\vec{v}$ instead of the binary expansion as described above.

5 Conclusion and Open Questions

We have presented a lattice-based predicate encryption scheme for inner product predicates whose security follows from the difficulty of the *learning with errors* (LWE) problem. Our construction can instantiate applications such as range and subset queries, polynomial evaluation, and CNF/DNF formulas on encrypted data. (A more detailed discussion of these applications appears in the full version of this paper [2].) Our construction is the first functional encryption scheme based on lattice techniques that goes beyond basic identity-based encryption.

Many open questions still remain in this field. One direction of research is to improve the security of our construction. Our scheme is weakly attribute hiding in the selective security model, but for stronger security guarantees we would like to construct a scheme that is fully secure and/or fully attribute hiding. Achieving either task will require

new simulation techniques; a natural question is whether the "dual-system" approach introduced by Waters [43] and used to prove full security of attribute-based encryption and predicate encryption in bilinear groups [29, 9, 33] can be adapted to lattice-based constructions.

Another direction of research is to improve the efficiency of our scheme. If $q = 2^{O(n)}$ is exponential size, as is needed for several of our applications, then setting the parameters as recommended in Section 4.4 gives public parameters of size $\Theta(\ell nm \lg^2(q)) = \Omega(\ell n^5)$ and ciphertexts of size $\Theta(\ell m \lg^2(q)) = \Omega(\ell n^4)$, which may be too large for practical purposes. A construction that achieved the same functionality with polynomial-size q would be a significant step forward. The *ring*-LWE problem introduced by Lyubashevsky, Peikert, and Regev [30] seems to be a natural candidate for such a construction.

Finally, it is a open question to construct predicate encryption schemes (via any technique) that support a greater range of functionality than inner product predicates. Ideally we would like a system that could support any polynomial-size predicate on encrypted data. Now that predicate encryption has moved into the world of lattices, perhaps techniques used to construct fully homomorphic encryption from lattices [23, 19, 18] could be used to help us move towards this goal.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. J. Cryptology 21(3), 350–391 (2008)
2. Agrawal, S., Freeman, D.M., Vaikuntanathan, V.: Functional encryption for inner product predicates from learning with errors. Cryptology ePrint Report 2011/410 (2011), `http://eprint.iacr.org`
3. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) Advances in Cryptology — EUROCRYPT 2010. Springer LNCS, vol. 6110, pp. 553–572 (2010), full version at `http://crypto.stanford.edu/~dabo/pubs/papers/latticebb.pdf`
4. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) Advances in Cryptology — CRYPTO '10. Springer LNCS, vol. 6223, pp. 98–115 (2010)
5. Agrawal, S., Boyen, X.: Identity-based encryption from lattices in the standard model. Manuscript (July 2009), `http://www.cs.stanford.edu/~xb/ab09/`
6. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) Automata, Languages, and Programming — ICALP '99. Springer LNCS, vol. 1644, pp. 1–9 (1999)
7. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: STACS. pp. 75–86 (2009), full version available at `http://www.cc.gatech.edu/~cpeikert/pubs/shorter.pdf`.
8. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute-based encryption. In: Shacham, H., Waters, B. (eds.) Pairing-Based Cryptography — Pairing '09. Springer LNCS, vol. 5671, pp. 248–265 (2009)

9. Attrapadung, N., Libert, B.: Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In: Nguyen, P.Q., Pointcheval, D. (eds.) Public Key Cryptography — PKC '10. Springer LNCS, vol. 6056, pp. 384–402 (2010)

10. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy. pp. 321–334 (2007)

11. Blundo, C., Iovino, V., Persiano, G.: Predicate encryption with partial public keys. In: Heng, S.H., Wright, R.N., Goi, B.M. (eds.) Cryptology and Network Security — CANS 2010. Springer LNCS, vol. 6467, pp. 298–313 (2010)

12. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology — EUROCRYPT '04. Springer LNCS, vol. 3027, pp. 223–238 (2004)

13. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) Advances in Cryptology — CRYPTO '04. Springer LNCS, vol. 3152, pp. 443–459 (2004)

14. Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology — EUROCRYPT '04. Springer LNCS, vol. 3027, pp. 506–522 (2004)

15. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. SIAM J. Comput. 32(3), 586–615 (2003), extended abstract in *CRYPTO '01*

16. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) Theory of Cryptography — TCC '11. Springer LNCS, vol. 6597, pp. 253–273 (2011)

17. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S. (ed.) Theory of Cryptography — TCC '07. Springer LNCS, vol. 4392, pp. 535–554 (2007)

18. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. To appear in *FOCS 2011* (2011), preprint available at `http://eprint.iacr.org/2011/344`

19. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) Advances in Cryptology — CRYPTO '11. Springer LNCS, vol. 6841, pp. 505–524 (2011)

20. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or, how to delegate a lattice basis. In: Gilbert, H. (ed.) Advances in Cryptology — EUROCRYPT '10. Springer LNCS, vol. 6110, pp. 523–552 (2010)

21. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S. (ed.) Theory of Cryptography — TCC '07. Springer LNCS, vol. 4392, pp. 515–534 (2007)

22. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) Advances in Cryptology — EUROCRYPT '06. Springer LNCS, vol. 4004, pp. 445–464 (2006)

23. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University (2009), `http://crypto.stanford.edu/craig`

24. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: 40th ACM Symposium on Theory of Computing — STOC '08. pp. 197–206. ACM (2008)

25. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) Advances in Cryptology — ASIACRYPT '02. Springer LNCS, vol. 2501, pp. 548–566 (2002)

26. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) Automata, Languages, and Programming — ICALP '08 (Part II). Springer LNCS, vol. 5126, pp. 579–591 (2008)

27. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security. pp. 89–98 (2006)

28. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N. (ed.) Eurocrypt 2008. Springer LNCS, vol. 4965, pp. 146–162 (2008), full version at `http://eprint.iacr.org/2007/404`

29. Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) Advances in Cryptology — EUROCRYPT '10. Springer LNCS, vol. 6110, pp. 62–91 (2010), full version available at `http://eprint.iacr.org/2010/110`

30. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) Advances in Cryptology — EUROCRYPT '10. Springer LNCS, vol. 6110, pp. 1–23 (2010)

31. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. In: 45th Annual IEEE Symposium on Foundations of Computer Science — FOCS '04. pp. 372–381 (2004)

32. Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: Matsui, M. (ed.) Advances in Cryptology — ASIACRYPT '09. Springer LNCS, vol. 5912, pp. 214–231 (2009)

33. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) Advances in Cryptology — CRYPTO '10. Springer LNCS, vol. 6223, pp. 191–208 (2010)

34. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security. pp. 195–203 (2007)

35. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: 41st Annual ACM Symposium on Theory of Computing — STOC '09. pp. 333–342 (2009)

36. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: 37th Annual ACM Symposium on Theory of Computing — STOC '05. pp. 84–93 (2005)

37. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) Advances in Cryptology — EUROCRYPT '05. Springer LNCS, vol. 3494, pp. 457–473 (2005)

38. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) Theory of Cryptography — TCC '09. Springer LNCS, vol. 5444, pp. 457–473 (2009)

39. Shi, E., Bethencourt, J., Chan, H.T.H., Song, D.X., Perrig, A.: Multi-dimensional range query over encrypted data. In: IEEE Symposium on Security and Privacy. pp. 350–364 (2007)

40. Shi, E., Waters, B.: Delegating capabilities in predicate encryption systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) Automata, Languages, and Programming — ICALP '08 (Part II). Springer LNCS, vol. 5126, pp. 560–578 (2008)

41. Shoup, V.: A Computational Introduction to Number Theory and Algebra, second edition. Cambridge University Press (2008)

42. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) Advances in Cryptology — EUROCRYPT '05. Springer LNCS, vol. 3494, pp. 114–127 (2005)

43. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) Advances in Cryptology — CRYPTO '09. Springer LNCS, vol. 5677, pp. 619–636 (2009)

44. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) Public Key Cryptography — PKC '11. Springer LNCS, vol. 6571, pp. 53–70 (2011)