Functional Encryption for Inner Product Predicates from Learning with Errors

Shweta Agrawal[*]
University of California, Los Angeles
shweta@cs.ucla.edu

David Mandell Freeman[†]
Stanford University
dfreeman@cs.stanford.edu

Vinod Vaikuntanathan[‡]
University of Toronto
vinodv@cs.toronto.edu

August 16, 2011

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).

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

[*] Part of this work done while at Microsoft Research Redmond. Research supported in part from a DARPA/ONR PROCEED award, and NSF grants 1118096, 1065276, 0916574 and 0830803.

[†] Research supported by NSF and DARPA.

[‡] Part of this work done while at Microsoft Research Redmond. Supported by an NSERC Discovery grant and by DARPA under Agreement number FA8750-11-2-0225. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

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* [15], 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) [16, 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 [16] as well as disjunctions, polynomial evaluation, and CNF and DNF formulas [28].

All known constructions of attribute-based encryption [37, 27, 9, 20, 34, 26, 44, 7, 29, 33, 8] and predicate encryption [13, 1, 39, 16, 28, 40, 38, 32, 10, 29] make use of bilinear groups, and the security of these schemes is based on many different, and often complex, assumptions. For example, in one assumption used by Katz, Sahai, and Waters [28, Assumption 1], 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 [16] and Katz, Sahai, and Waters [28]. 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] 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 groups [14, 25, 11, 12, 42, 22], our construction builds on the ideas and techniques used to achieve identity-based encryption from the LWE assumption [24, 4, 19, 2, 3]. 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 [2]. 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 [2]. 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 and a vector $\mathbf{x}$, and the secret key is a short vector $\mathbf{s}$ in the coset of the dual lattice $\Lambda^\perp$ defined by $\mathbf{x}$. A ciphertext contains a vector $\mathbf{y}$ that is "close" to a random vector $\mathbf{r} \in \Lambda$, so taking the inner product of $\mathbf{y}$ with $\mathbf{s}$ and reducing mod $\mathbb{Z}$ gives a value that is "close" to $\langle \mathbf{x}, \mathbf{r} \rangle \bmod \mathbb{Z}$. The term $\langle \mathbf{x}, \mathbf{r} \rangle \bmod \mathbb{Z}$ can thus be used (with some additional "noise") as a one-time pad to encrypt a message bit.

Existing constructions of lattice-based IBE in the standard model [4, 19, 2, 3] 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 lattice match ciphertext lattices in the same way as above.

Before explaining our solution to this problem, let us recall the IBE scheme of Agrawal, Boneh, and Boyen [2]. 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 [4, 19]. In the 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

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

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

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

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 $T_{\vec{v}} : (\mathbb{Z}_q^m)^{\ell+1} \to \mathbb{Z}_q^{2m}$ 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

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

Letting the secret key for $\vec{v}$ be the GPV secret key associated to $\Lambda_q^\perp(\mathbf{A}_0 \ \| \ \sum_{i=1}^\ell v_i \mathbf{A}_i)$ allows the holder of a key for predicate $\vec{v}$ to decrypt a ciphertext associated with attribute $\vec{w}$ exactly when $\langle \vec{v}, \vec{w}\rangle = 0$.

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 r-ary decomposition of the vector $\vec{v}$ for suitably small r (e.g., $r = 2$), at the expense of expanding the ciphertext by a factor of $\log_r q$.

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 [16]. 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 $\perp$.

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 $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{sk}, f)$, 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 $\mathsf{Dec}(\mathsf{sk}_f, C) = \perp$ 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. 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 Appendix A.

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 let $\mathbf{Id}_m$ denote the $m \times m$ identity matrix. 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.

If $\mathbf{A}_1$ is an $n \times m$ matrix and $\mathbf{A}_2$ is an $n \times m'$ matrix, then $[\mathbf{A}_1 \| \mathbf{A}_2]$ denotes the $n \times (m + m')$ matrix formed by concatenating $\mathbf{A}_1$ and $\mathbf{A}_2$. If $\mathbf{x}_1$ is a length m vector and $\mathbf{x}_2$ is a length m' vector, then we let $[\mathbf{x}_1 | \mathbf{x}_2]$ denote the length $(m + m')$ vector formed by concatenating $\mathbf{x}_1$ and $\mathbf{x}_2$. However, when doing matrix-vector multiplication we always view vectors as column vectors.

We say a function $f(n)$ is *negligible* if it is $O(n^{-c})$ for all $c > 0$, and we use $\mathrm{negl}(n)$ to denote a negligible function of n. We say $f(n)$ is *polynomial* if it is $O(n^c)$ for some $c > 0$, and we use $\mathrm{poly}(n)$ to denote a polynomial function of n. We say an event occurs with *overwhelming probability* if its probability is $1 - \mathrm{negl}(n)$. The function $\lg x$ is the base 2 logarithm of x. The notation $\lfloor x \rceil$ denotes the nearest integer to x, rounding towards 0 for half-integers.

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

$$\begin{aligned}
\Lambda_q^{\perp}(\mathbf{A}) &:= \left\{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{e} = \mathbf{0} \bmod q\right\} \\
\Lambda_q^{\mathbf{u}}(\mathbf{A}) &:= \left\{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q\right\}. \\
\Lambda_q(\mathbf{A}) &:= \left\{\mathbf{e} \in \mathbb{Z}^m : \exists\, \mathbf{s} \in \mathbb{Z}_q^m \text{ with } \mathbf{A}^t \cdot \mathbf{s} = \mathbf{e} \bmod q\right\}.
\end{aligned}$$

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 [5] and later Alwen and Peikert [6] showed how to sample an essentially 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 ([6, 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 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*

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

with overwhelming probability in n.

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*

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

3.2 Sampling algorithms

We will also need the following algorithms to sample short vectors from specific lattices. Looking ahead, the algorithm $\mathsf{SampleLeft}$ [2, 19] will be used to sample keys in the real system, while the algorithm $\mathsf{SampleRight}$ [2] will be used to sample keys in the simulation.

Algorithm $\mathsf{SampleLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T}_A, \mathbf{u}, \sigma)$

Inputs: a full rank matrix $\mathbf{A}$ in $\mathbb{Z}_q^{n\times m}$ and a "short" basis $\mathbf{T}_A$ of $\Lambda_q^{\perp}(\mathbf{A})$,
a matrix $\mathbf{B}$ in $\mathbb{Z}_q^{n\times m_1}$, (3.1)
a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter σ.

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 ([2, Theorem 17], [19, Lemma 3.2])**.** *Let $q > 2,\ m > n$ and $\sigma > \|\widetilde{\mathbf{T}_A}\| \cdot \omega(\sqrt{\log(m+m_1)})$. Then $\mathsf{SampleLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T}_A, \mathbf{u}, \sigma)$ 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 $\mathsf{SampleRight}(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T}_{\mathbf{B}}, \mathbf{u}, \sigma)$

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 $\mathbb{Z}_q^{n\times m}$ and a "short" basis $\mathbf{T}_B$ of $\Lambda_q^{\perp}(\mathbf{B})$, (3.2)
a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter σ.

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

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-$\mathsf{LWE}_{q,n,\chi}$ 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*-$\mathsf{LWE}_{q,n,\chi}$ *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-$\mathsf{LWE}_{q,n,\chi}$ 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-$\mathsf{LWE}_{q,n,\overline{\Psi}_\alpha}$, 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-$\mathsf{LWE}_{q,n,\overline{\Psi}_\alpha}$, 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 ([2, 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. Let $r = r(n, \ell) \geq 2$ be an integer and define $k = k(n, \ell) := \lfloor \log_r q \rfloor$.

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 $i = 1, \dots, \ell$ and $\gamma = 0, \dots, k$.
3. Select a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$.

Output $\mathsf{PP} = (\mathbf{A}, \{\mathbf{A}_{i,\gamma}\}_{i \in \{1,\dots,\ell\}, \gamma \in \{0,\dots,k\}}, \mathbf{u})$ 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 $\vec{v} = (v_1, \dots, v_\ell) \in \mathbb{Z}_q^\ell$, do:

1. For $i = 1, \dots, \ell$, let $\hat{v}_i$ be the integer in $[0, q-1]$ congruent to $v_i \bmod q$. Write the r-ary decomposition of $\hat{v}_i$ as
$$\hat{v}_i = \sum_{\gamma=0}^{k} v_{i,\gamma} \cdot r^\gamma, \tag{4.1}$$
where $v_{i,\gamma}$ are integers in $[0, r-1]$.
2. Define the matrices
$$\begin{aligned} \mathbf{C}_{\vec{v}} &:= \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{A}_{i,\gamma} && \in \mathbb{Z}_q^{n \times m} \\ \mathbf{A}_{\vec{v}} &:= [\mathbf{A} \parallel \mathbf{C}_{\vec{v}}] && \in \mathbb{Z}_q^{n \times 2m}. \end{aligned}$$
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 $\mathbf{A}_{\vec{v}} \cdot \mathbf{e} = \mathbf{u} \mod q$.

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 $\mathbf{B} \overset{\text{R}}{\leftarrow} \mathbb{Z}_q^{n \times m}$.
2. Choose a uniformly random $\mathbf{s} \overset{\text{R}}{\leftarrow} \mathbb{Z}_q^n$.
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 $i = 1, \dots, \ell$ and $\gamma = 0, \dots, k$, do the following:
   (a) Pick a random matrix $\mathbf{R}_{i,\gamma} \in \{-1, 1\}^{m \times m}$.
   (b) Compute $\mathbf{c}_{i,\gamma} \leftarrow (\mathbf{A}_{i,\gamma} + r^\gamma w_i \mathbf{B})^\intercal \mathbf{s} + \mathbf{R}_{i,\gamma}^\intercal \mathbf{x}$.
6. Compute $c' \leftarrow \mathbf{u}^\intercal \mathbf{s} + x + M \cdot \lfloor q/2 \rceil \in \mathbb{Z}_q$.

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

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

1. Define the r-ary 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 $z \leftarrow c' - \mathbf{e}^{\mathsf{T}} \mathbf{c} \pmod q$.

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

Note that, unlike in [2], the only role played by the matrix $\mathbf{B}$ in the main system is that of a "ciphertext polluter." Concretely, this means that the terms involving $\mathbf{B}$ vanish, and thus enable decryption, exactly when the conditions are right; i.e., when $\langle \vec{v}, \vec{w} \rangle = 0$. Thus, the encryptor can choose, use, and discard this matrix ephemerally and the key generation procedure does not need $\mathbf{B}$ at all. This is not the case in the IBE of [2], where $\mathbf{B}$ is crucially required by *both* the encryption and key generation algorithms.

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 $\mathsf{LinFE.Dec}$ algorithm recovers M.

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

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

Let

$$\mathbf{e} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, \vec{v}), \quad \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{PP}, \vec{w}, M), \quad \textit{and} \quad \tilde{M} \leftarrow \mathsf{Dec}(\mathsf{PP}, \mathbf{e}, \mathsf{CT}).$$

If $\langle \vec{v}, \vec{w} \rangle = 0 \pmod q$, then with overwhelming probability we have $M' = M$.

Proof. During the first step of $\mathsf{LinFE.Dec}$, 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

$$\begin{aligned}\mathbf{c}_{\vec{v}} &= \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \left[(\mathbf{A}_{i,\gamma} + r^{\gamma} w_i \mathbf{B})^{\mathsf{T}} \mathbf{s} + \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x}\right] \\ &= \left(\sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{A}_{i,\gamma}\right)^{\mathsf{T}} \mathbf{s} + \underbrace{\left(\sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} r^{\gamma} v_{i,\gamma} w_i\right)}_{\vec{v} \cdot \vec{w}} \mathbf{B}^{\mathsf{T}} \mathbf{s} + \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x}. \end{aligned} \qquad (4.2)$$

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 $\mathsf{LinFE.Dec}$, we have:

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

In the third step of $\mathsf{LinFE.Dec}$, 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

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

Finally, we compute:

$$\begin{aligned} z &= c' - \mathbf{e}^{\mathsf{T}} \mathbf{c} \pmod q \\ &= (\mathbf{u}^{\mathsf{T}} \mathbf{s} + x + M \cdot \lfloor q/2 \rceil) - \mathbf{u}^{\mathsf{T}} \mathbf{s} - \mathbf{e}^{\mathsf{T}} \left[\mathbf{x} \middle| \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \pmod q \\ &= M \cdot \lfloor q/2 \rceil + \underbrace{\left(x - \mathbf{e}^{\mathsf{T}} \left[\mathbf{x} \middle| \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \right)}_{\text{low-norm noise}} \pmod q \end{aligned}$$

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

$$\left| x - \mathbf{e}^{\mathsf{T}} \left[\mathbf{x} \middle| \sum_{i=1}^{\ell} \sum_{\gamma=0}^{k} v_{i,\gamma} \mathbf{R}_{i,\gamma}^{\mathsf{T}} \mathbf{x} \right] \right| < q/4. \tag{4.3}$$

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

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

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

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

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

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

(Recall that $k = \lfloor \log_r q \rfloor$.) 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. □

4.3 Security

We use the simulation technique of Agrawal, Boneh, and Boyen [2] 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 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. The first and last games correspond to the real security game with challenge ciphertexts $\mathsf{LinFE.Enc}(\mathsf{PP}, \vec{w}_0, M_0)$ and $\mathsf{LinFE.Enc}(\mathsf{PP}, \vec{w}_1, M_1)$, respectively. In the intermediate games we use "alternative" setup, key generation, and encryption algorithms Sim.Setup, Sim.KeyGen, and Sim.Enc. The algorithm Sim.Setup takes as additional input an attribute vector $\vec{w}^*$, and Sim.Enc takes as additional input the master key output by 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 $\langle \vec{v}, \vec{w}_1 \rangle \neq 0$.

Game$_0$: The challenger runs the LinFE.Setup algorithm, answers the adversary's secret key queries using the LinFE.KeyGen algorithm, and generates the challenge ciphertext using the LinFE.Enc algorithm with attribute $\vec{w}_0$ and message M_0.

Game$_1$: The challenger runs the Sim.Setup algorithm with $\vec{w}^* = \vec{w}_0$ and answers the adversary's secret key queries using the Sim.KeyGen algorithm. The challenger generates the challenge ciphertext using the Sim.Enc algorithm with attribute $\vec{w}_0$ and message M_0.

Game$_2$: The challenger runs the Sim.Setup algorithm with $\vec{w}^* = \vec{w}_0$ and answers the adversary's secret key queries using the Sim.KeyGen algorithm. The challenger generates the challenge ciphertext by choosing a uniformly random element of the ciphertext space.

Game$_3$: The challenger runs the Sim.Setup algorithm with $\vec{w}^* = \vec{w}_1$ and answers the adversary's secret key queries using the Sim.KeyGen algorithm. The challenger generates the challenge ciphertext by choosing a uniformly random element of the ciphertext space.

Game$_4$: The challenger runs the Sim.Setup algorithm with $\vec{w}^* = \vec{w}_1$ and answers the adversary's secret key queries using the Sim.KeyGen algorithm. The challenger generates the challenge ciphertext using the Sim.Enc algorithm with attribute $\vec{w}_1$ and message M_1.

Game$_5$: The challenger runs the LinFE.Setup algorithm, answers the adversary's secret key queries using the LinFE.KeyGen algorithm, and generates the challenge ciphertext using the LinFE.Enc algorithm with attribute $\vec{w}_1$ and message M_1.

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 $i = 1, \dots, \ell$ and $\gamma = 0, \dots, k$, pick random matrices $\mathbf{R}^*_{i,\gamma} \xleftarrow{\mathrm{R}} \{-1, 1\}^{m\times m}$ and set
$$\mathbf{A}_{i,\gamma} \leftarrow \mathbf{A}\,\mathbf{R}^*_{i,\gamma} - r^\gamma w_i^*\,\mathbf{B}^*.$$

Output the public parameters and master key

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

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
$$\begin{aligned} \mathbf{C}_{\vec{v}} &:= \sum_{i=1}^{\ell}\sum_{\gamma=0}^{k} v_{i,\gamma}\mathbf{A}_{i,\gamma} \qquad \in \mathbb{Z}_q^{n\times m} \\ \mathbf{A}_{\vec{v}} &:= [\mathbf{A} \,\|\, \mathbf{C}_{\vec{v}}] \qquad \in \mathbb{Z}_q^{n\times 2m}. \end{aligned}$$
Observe that
$$\mathbf{A}_{\vec{v}} = \left[\mathbf{A} \,\middle\|\, \mathbf{A}\Big(\sum_{i=1}^{\ell}\sum_{\gamma=0}^{k} v_{i,\gamma}\mathbf{R}^*_{i,\gamma}\Big) - \underbrace{\Big(\sum_{i=1}^{\ell}\sum_{\gamma=0}^{k} r^\gamma v_{i,\gamma} w_i^*\Big)}_{\langle\vec{v},\vec{w}^*\rangle}\mathbf{B}^*\right].$$
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 $\mathbf{B}$.
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 $i = 1, \dots, \ell$ and $\gamma = 0, \dots, k$.

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.

(a) The view of the adversary $\mathcal{A}$ in Game_0 *is statistically close to the view of $\mathcal{A}$ in* Game_1.

(b) The view of the adversary $\mathcal{A}$ in Game_4 *is statistically close to the view of $\mathcal{A}$ in* Game_5.

Proof. It suffices to prove (a) only. We first show that public parameters and challenge ciphertext output by the alternative Sim.Setup and Sim.Enc algorithms are statistically indistinguishable from those output by the real Setup and Enc algorithms.

First, the matrix $\mathbf{A}$ in the public parameters is chosen by running TrapGen in Game_0, whereas it is a uniformly random matrix in $\mathbb{Z}_q^{n\times m}$ in Game_1. Since $m \geq 6n\log q$, by Theorem 3.1, the matrix $\mathbf{A}$ output by TrapGen is statistically indistinguishable from a uniformly random matrix, and thus the distribution of $\mathbf{A}$ in Game_0 and Game_1 are statistically close.

Next, we show that the joint distribution of $\mathbf{A}_{i,\gamma}$ in the public parameters and $\mathbf{c}_{i,\gamma}$ in the ciphertext in Game_0 and Game_1 are statistically indistinguishable, for every $i \in \{1,\dots,\ell\}$ and $\gamma \in \{0,\dots,k\}$. The difference between $(\mathbf{A}_{i,\gamma},\mathbf{c}_{i,\gamma})$ in the two games is as follows:

- In Game_0 the matrix $\mathbf{A}_{i,\gamma}$ is uniformly random in $\mathbb{Z}_q^{n\times m}$ for every $i \in \{1,\dots,\ell\}$ and $\gamma \in \{0,\dots,k\}$. In Game_1, $\mathbf{A}_{i,\gamma}$ is equal to $\mathbf{A}\mathbf{R}^*_{i,\gamma} - r^\gamma w_i^*\mathbf{B}^*$ where (by Theorem 3.1) $\mathbf{B}^*$ is statistically close to uniformly random in $\mathbb{Z}_q^{n\times m}$ (and is the same for all i,γ), and the matrices $\mathbf{R}^*_{i,\gamma}$ are independently chosen from $\{-1,1\}^{m\times m}$ for every i and γ.

- In Game_0 the challenge ciphertext components $\mathbf{c}_{i,\gamma}$ are computed as

$$\mathbf{c}_{i,\gamma} = \left(\mathbf{A}_{i,\gamma} + r^\gamma w_i^*\mathbf{B}^*\right)^{\mathsf{T}}\mathbf{s} + {\mathbf{R}^*_{i,\gamma}}^{\mathsf{T}}\mathbf{x},$$

  where $\mathbf{B}^*$ is uniformly random in $\mathbb{Z}_q^{n\times m}$ (and is the same for all i,γ) and the matrices $\mathbf{R}^*_{i,\gamma}$ are independently chosen from $\{-1,1\}^{m\times m}$ for every i and γ.

  In contrast, in Game_1, the corresponding challenge ciphertext components are

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

  where $\mathbf{R}^*_{i,\gamma}$ are the same matrices used to compute the public parameters $\mathbf{A}_{i,\gamma}$. Indeed, the main difference between the two games is that the matrices $\mathbf{R}^*_{i,\gamma}$ are chosen by the encryption algorithm and used only in the ciphertext $\mathbf{c}_{i,\gamma}$ in Game_0, whereas in Game_1, they play a double role: they are used to construct the matrices $\mathbf{A}_{i,\gamma}$ in the public parameters as well as the ciphertext $\mathbf{c}_{i,\gamma}$.

We now argue that the distributions of the set $\mathcal{S} := \left(\mathbf{A}, \{\mathbf{A}_{i,\gamma},\mathbf{c}_{i,\gamma}\}_{i\in\{1,\dots,\ell\},\gamma\in\{0,\dots,k\}}\right)$ in Game_0 and Game_1 are statistically indistinguishable. The key is to observe that for every i and γ, if $\mathbf{A}_{i,\gamma}$ is uniformly random and $\mathbf{R}^*_{i,\gamma}$ is uniformly random in $\{-1,1\}^{m\times m}$, then it follows from Lemma A.2 that the following two distributions are statistically indistinguishable for every fixed matrix $\mathbf{B}^*$, every $\vec{w}^*$ and every vector $\mathbf{x} \in \mathbb{Z}_q^m$:

$$(\mathbf{A}, \mathbf{A}\mathbf{R}^*_{i,\gamma} - r^\gamma w_i^*\mathbf{B}^*, {\mathbf{R}^*_{i,\gamma}}^{\mathsf{T}}\mathbf{x}) \approx_s (\mathbf{A}, \mathbf{A}_{i,\gamma}, {\mathbf{R}^*_{i,\gamma}}^{\mathsf{T}}\mathbf{x}) \tag{4.5}$$

Furthermore, since the matrices $\mathbf{R}^*_{i,\gamma}$ are chosen independently for every i, γ, the joint distributions of these quantities for all i, γ are also statistically close:

$$\left(\mathbf{A}, \left\{\mathbf{A}\mathbf{R}^*_{i,\gamma} - r^\gamma w_i^* \mathbf{B}^*, {\mathbf{R}^*_{i,\gamma}}^\mathsf{T}\mathbf{x}\right\}_{i,\gamma}\right) \approx_s \left(\mathbf{A}, \left\{\mathbf{A}_{i,\gamma}, {\mathbf{R}^*_{i,\gamma}}^\mathsf{T}\mathbf{x}\right\}_{i,\gamma}\right) \tag{4.6}$$

If we add the same quantity to both sides of Equation 4.6 and use the fact that that applying any function to two statistically indistinguishable ensembles produces statistically indistinguishable ensembles, we see that the following two distributions are statistically close:

$$\begin{aligned}&\left(\mathbf{A}, \left\{\mathbf{A}\mathbf{R}^*_{i,\gamma} - r^\gamma w_i^* \mathbf{B}^*, \underbrace{\left(\mathbf{A}\mathbf{R}^*_{i,\gamma} - r^\gamma w_i^* \mathbf{B}^* + r^\gamma w_i^* \mathbf{B}^*\right)^\mathsf{T}\mathbf{s}}_{\text{added term}} + {\mathbf{R}^*_{i,\gamma}}^\mathsf{T}\mathbf{x}\right\}_{i,\gamma}\right) \\ &\qquad\qquad\qquad \approx_s \left(\mathbf{A}, \left\{\mathbf{A}_{i,\gamma}, \underbrace{\left(\mathbf{A}_{i,\gamma} + r^\gamma w_i^* \mathbf{B}^*\right)^\mathsf{T}\mathbf{s}}_{\text{added term}} + {\mathbf{R}^*_{i,\gamma}}^\mathsf{T}\mathbf{x}\right\}_{i,\gamma}\right)\end{aligned} \tag{4.7}$$

Now observe that the distribution on the left hand side of (4.7) is the distribution of the public parameters and the challenge ciphertext in Game_1, while that on the right hand side is the distribution in Game_0. It follows that the joint distributions of the public parameters and the challenge ciphertext in the two games are statistically indistinguishable.

To conclude the proof, we show that the secret keys output by $\mathsf{Sim.KeyGen}$ are statistically indistinguishable from those output by $\mathsf{LinFE.KeyGen}$ (given the public parameters and the challenge ciphertext). Assuming σ is sufficiently large, this follows from the properties of the algorithms SampleLeft and SampleRight. By the properties of SampleLeft (Theorem 3.3), the key $\mathsf{sk}_{\vec{v}}$ in Game_0 comes from a distribution statistically indistinguishable from $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}_{\vec{v}}),\sigma}$. By the properties of SampleRight (Theorem 3.4), the key $\mathsf{sk}_{\vec{v}}$ in Game_1 comes from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}_{\vec{v}}),\sigma}$ as well. In Section 4.4 we will analyze exactly how large σ must be for these statements to hold.

□

Lemma 4.4.

*(a) If the decision-*LWE *assumption holds, then the view of the adversary $\mathcal{A}$ in* Game_1 *is computationally indistinguishable from the view of $\mathcal{A}$ in* Game_2.

*(b) If the decision-*LWE *assumption holds, then the view of the adversary $\mathcal{A}$ in* Game_3 *is computationally indistinguishable from the view of $\mathcal{A}$ in* Game_4.

Proof. It suffices to prove (a) only. Suppose we are given $m + 1$ LWE challenges $(\mathbf{a}_i, y_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for $j = 0, \dots, m$, where either $y_j = \langle \mathbf{a}_j, \mathbf{s} \rangle + x_j$ for some (fixed) random secret $\mathbf{s} \overset{R}{\leftarrow} \mathbb{Z}_q^n$ 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:

$$\begin{array}{ll} \mathbf{A} := \begin{pmatrix} | & & | \\ \mathbf{a}_1 & \cdots & \mathbf{a}_m \\ | & & | \end{pmatrix} \in \mathbb{Z}_q^{n \times m} & \qquad \mathbf{u} := \mathbf{a}_0 \\ \mathbf{c}_0 := \ (y_1, \dots, y_m) \ \in \mathbb{Z}_q^m & \qquad c' := y_0 + M \cdot \lfloor \frac{q}{2} \rceil \end{array} \tag{4.8}$$

We simulate the challenger as follows:

- **Setup:** Run $\mathsf{Sim.Setup}$ with $\vec{w}^* = \vec{w}_0$, and let $\mathbf{A}$ and $\mathbf{u}$ be as in (4.8).
- **Private key queries:** Run the $\mathsf{Sim.KeyGen}$ 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 $(\mathbf{c}_0, \{\mathbf{c}_{i,\gamma}\}, c')$.

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

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

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 a Game_1 challenger.

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 to a Game_2 challenger.

We conclude that any efficient adversary that can distinguish Game_1 from Game_2 can solve the decision-LWE problem. □

Lemma 4.5. *The view of the adversary $\mathcal{A}$ in Game_2 is statistically indistinguishable from the view of $\mathcal{A}$ in Game_3.*

Proof. Note that the only place where $\vec{w}^*$ appears in the two games is in the public parameter $\mathbf{A}_{i,\gamma} := \mathbf{A}\mathbf{R}^*_{i,\gamma} - r^\gamma w_i^* \mathbf{B}^*$. Let $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n \times \ell(k+1)}$ and $\overline{\mathbf{R}}^* \in \mathbb{Z}_q^{\ell(k+1) \times \ell(k+1)}$ 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 A.2, $(\mathbf{A}, \mathbf{A}\overline{\mathbf{R}}^*)$ 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. □

Proof of Theorem 4.2. Suppose that there is an efficient adversary $\mathcal{A}$ that wins the security game. Let $\mathcal{A}^{(i)}$ denote the output of $\mathcal{A}$ interacting with Game_i. Then we have

$$\left|\Pr[\mathcal{A}^{(0)} = 1] - \Pr[\mathcal{A}^{(5)} = 1]\right| \geq \frac{1}{\mathrm{poly}(n)}.$$

By a standard hybrid argument, this implies that

$$\left|\Pr[\mathcal{A}^{(i)} = 1] - \Pr[\mathcal{A}^{(i+1)} = 1]\right| \geq \frac{1}{\mathrm{poly}(n)} \tag{4.9}$$

for some i in $0, \ldots, 4$. Since $\mathcal{A}$ is polynomial time, Lemma 4.3 implies that (4.9) cannot hold for $i = 0$ or $i = 4$, while Lemma 4.5 implies that (4.9) cannot hold for $i = 2$. It now follows from Lemma 4.4 that $\mathcal{A}$ can be used to solve the decision-LWE problem. □

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

$$\frac{q}{\log q} = \Omega\left(\sigma \cdot \ell \cdot \frac{r}{\log r} \cdot m^{3/2}\right) \qquad \text{and} \qquad \alpha \le \left(\log q \cdot \sigma \cdot \ell \cdot \frac{r}{\log r} \cdot m \cdot \omega\sqrt{\log m}\right)^{-1}. \tag{4.10}$$

In our security theorem (Theorem 4.2), we require

$$m > 6n \lg q \tag{4.11}$$

in order for the output of TrapGen to be statistically random. The additional constraints imposed by our security reduction are the following:

- From the description of LinFE.Setup and LinFE.KeyGen, we have $\|\widetilde{\mathbf{T}_{\mathbf{A}}}\| = O(\sqrt{n \log q})$ (by Theorem 3.1) and $\mathbf{e} \leftarrow D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}||\mathbf{B}),\sigma}$ (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 Sim.Setup and Sim.KeyGen, we have $\|\widetilde{\mathbf{T}_{\mathbf{B}^*}}\| = O(\sqrt{n \log q})$ (by Theorem 3.1), and $\mathbf{e} \leftarrow D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}||\mathbf{B}),\sigma}$ where, by Theorem 3.4, subject to the requirement that
$$\sigma \ge \|\widetilde{\mathbf{T}}_{\mathbf{B}^*}\| \cdot s_R \cdot \omega(\sqrt{\log m}) \tag{4.12}$$
Since $\mathbf{R}$ is a sum of $\ell \cdot (\log_r q + 1)$ random matrices with $\{1, -1\}$ entries, it follows from Lemma A.1 that $s_R = \sup_{\{\mathbf{x}:\|\mathbf{x}\|=1\}} \|\mathbf{R}\mathbf{x}\| = O(\ell \cdot (\log_r q + 1) \cdot \sqrt{m})$ with overwhelming probability. Plugging this value into (4.12), we see that it suffices to choose
$$\sigma \ge O(\sqrt{n \log q}) \cdot O(\ell \cdot (\log_r q + 1) \cdot \sqrt{m}) \cdot \omega(\sqrt{\log m}).$$

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

$$\sigma = \omega(m\ell \log q \cdot \sqrt{\log m}), \tag{4.13}$$

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

In order to reduce decision-LWE to approximating worst-case lattice problems to within $\mathrm{poly}(n)$ 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 $\alpha \ge 1/\mathrm{poly}(n)$, while for exponential size q we can use Peikert's classical reduction (Theorem 3.8) with each prime factor q_i of q satisfying $\omega(\sqrt{\log n})/\alpha < q_i < \mathrm{poly}(n)$. (Note that a large value of q may be required for certain applications; see Section 5.)

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

$$\begin{array}{rcll}
r & = & 2 & \\
m & = & \lceil n^{1+\delta} \rceil, & \text{to satisfy (4.11)} \\
\sigma & = & \lceil n^{2+2\delta} \cdot \ell \rceil, & \text{to satisfy (4.13)} \\
q_i & = & \text{the } i\text{th prime larger than } (\ell \log \ell)^2 \cdot n^{7/2+5\delta} & \\
\alpha & = & \Omega\left((\ell \log \ell)^2 \cdot n^{3+5\delta}\right)^{-1} & \text{to satisfy (4.10)}
\end{array}$$

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 $\mathsf{SampleLeft}$ and $\mathsf{SampleRight}$ algorithms.

Finally, we note that one might be able to choose these parameters to have somewhat smaller values, however we have made no attempt to optimize them. In particular, one might be able to reduce the ciphertext size by choosing a larger value of r.

4.5 Multi-bit Encryption

As in the case of the GPV encryption scheme [24, §7] and the ABB IBE scheme [2, §6.5], our scheme can be extended to encrypt multiple bits simultaneously using the same encryption randomness $\mathbf{s}$. Briefly, we can do this by replacing the public parameter $\mathbf{u}$ with one vector $\mathbf{u}_j \in \mathbb{Z}_q^n$ for each message bit and encrypting the ith message bit M_j using $\mathbf{u}_j$ to form the ciphertext component

$$c'_j \leftarrow \mathbf{u}_j^{\mathsf{T}} \mathbf{s} + x_j + M_j \cdot \lfloor q/2 \rceil \in \mathbb{Z}_q.$$

(where $x_j \leftarrow \overline{\Psi}_\alpha$). The components $\mathbf{c}_0$ and $\mathbf{c}_{i,\gamma}$ remain as before for all i, γ. The security analysis remains almost exactly the same, except that in the reduction from LWE in Lemma 4.4 we use m LWE samples to produce the matrix $\mathbf{A}$ and the ciphertext component $\mathbf{c}_0$, and use one additional sample for each message bit to produce $\mathbf{u}_j$ and c'_j.

This multi-bit version of our scheme allows us to achieve the semantics of Definition 2.1 as discussed on page 4. In particular, our proof of security shows that for $\mathsf{CT} \leftarrow \mathsf{LinFE.Enc}(\mathsf{PP}, \vec{w}, M)$ and $\mathsf{sk}_{\vec{v}} \leftarrow \mathsf{LinFE.KeyGen}(\mathsf{MK}, \vec{v})$, if $\langle \vec{v}, \vec{w} \rangle \neq 0$ then $\mathsf{LinFE.Dec}(\mathsf{sk}_v, \mathsf{CT})$ is indistinguishable from a random bit under the decision-LWE assumption.

5 Applications

Katz, Sahai, and Waters [28, §5] describe a number of applications of their predicate encryption scheme for inner product predicates. Predicate and attribute vectors in their scheme are defined over $\mathbb{Z}_N$ for some N that is exponential in the security parameter n, and attributes and predicates can correspond to any vector in $\mathbb{Z}_N^\ell$. In our scheme vectors are defined over $\mathbb{Z}_q$ where q may be either polynomial in n or exponential in n; attributes and predicates now correspond to any vector in $\mathbb{Z}_q^\ell$.

The key technique that allows us to handle arbitrary predicate vectors $\vec{v} \in \mathbb{Z}_q^\ell$ is the decomposition of $\vec{v}$ into its r-ary representation for some small r (e.g., $r = 2$). As a consequence, if it is known in advance that predicate vectors will have only entries in $[0, r-1]$ (for example, if the vector takes only binary values), then all of the matrices $\mathbf{A}_{i,\gamma}$ and ciphertext terms $\mathbf{c}_{i,\gamma}$ may be discarded for $\tau \geq 1$, thus reducing the size of both the public parameters and the ciphertext by a factor of approximately $k = \log_r q$.

We now consider the principal applications described by Katz, Sahai, and Waters and determine what effect, if any, the size of q and the r-ary decomposition of predicate vectors have on each.

Hidden vector encryption. Boneh and Waters [16] developed a predicate encryption scheme called *hidden vector encryption* (HVE) and showed how the scheme can be used to perform conjunctions of subset and comparison queries. Katz, Sahai, and Waters [28, §5.2] showed how HVE can be realized using inner product predicates.

Briefly, let Σ be a set and $\Sigma_\star = \Sigma \cup \{\star\}$. For vectors $\vec{a} \in \Sigma_*^k$ and $\vec{x} \in \Sigma^k$, define

$$\phi_{\vec{a}}^{\mathsf{hve}}(\vec{x}) = \begin{cases} 1 & \text{if for all } i \text{ either } a_i = x_i \text{ or } a_i = \star. \\ 0 & \text{otherwise.} \end{cases}$$

To encode this predicate using inner products, we assume $\Sigma \subset \mathbb{Z}_q \setminus \{0\}$ and associate to the predicate $\vec{a}$ a vector $\vec{A} \in \mathbb{Z}_q^{2k}$ with $A_{2i-1} = 1$ and $A_{2i} = a_i$ if $a_i \neq \star$, and $A_{2i-1} = A_{2i} = 0$ if $a_i = \star$. For the attribute vector $\vec{x}$ we choose a random vector $\vec{r} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^k$ and associate to $\vec{x}$ the vector $\vec{X}$ with $X_{2i-1} = -r_i x_i$ and $X_{2i} = r_i$.

Correctness of this encoding requires that a random element of $\mathbb{Z}_q$ be zero with negligible probability, which implies that q must be superpolynomial. We must therefore use exponential size q and Peikert's worst-case reduction (Theorem 3.8). However, since the random elements r_i appear in attribute vectors only, if predicate vectors contain only small entries then we can reduce the size of the public parameters and ciphertext by eliminating the $\mathbf{A}_{i,\gamma}$ and $\mathbf{c}_{i,\gamma}$ for $\gamma \geq 1$. Indeed, in the applications of HVE described by Boneh and Waters [16, §6], the predicate vectors have entries in $\{0, 1\}$.

Polynomial evaluation and CNF/DNF formulae. Katz, Sahai, and Waters [28, §5.3] observe that inner products of length k vectors can be used to evaluate degree d polynomials in t variables as long as $k > t^d$. In this formulation, predicate vectors encode the coefficients of a polynomial f and attribute vectors encode all monomials in the variables. The predicate is 1 if and only if f evaluates to zero.

Our scheme supports both the polynomial evaluation functionality (for any q) as well as the "dual" concept, where attributes are coefficients and predicates are monomials. Furthermore, if the polynomial coefficients are small, then we can compress the public parameters and ciphertext as discussed above. (The same holds if the variables in the "dual" system take only small values.)

Katz, Sahai, and Waters also show how to use polynomial evaluation to evaluate conjunctions and disjunctions: to test whether $x_1 = a_1$ or $x_2 = a_2$ we evaluate the polynomial $(x_1 - a_1) \cdot (x_2 - a_2)$. Assuming the variables take values in a small set, this predicate can be handled by our system using polynomial size q.

On the other hand, to test whether $x_1 = a_1$ and $x_2 = a_2$ we evaluate the polynomial $r_1(x_1 - a_1) + r_2(x_2 - a_2)$ for random r_1, r_2. As in the HVE example, correctness depends on a random element of $\mathbb{Z}_q$ being zero with negligible probability. Thus to test conjunctions we require q to be exponential.

We conclude that when q is of exponential size we can use our scheme to implement either key-policy or ciphertext-policy attribute based encryption [27, 9] where policies are given by CNF or DNF formulae. Furthermore, if implementing ciphertext-policy ABE with boolean attribute variables, then we can compress the public parameters and ciphertexts by removing $\mathbf{A}_{i,\gamma}$ and $\mathbf{c}_{i,\gamma}$ for $\gamma \geq 1$.

Remark 5.1. It was pointed out to us by Brent Waters that predicates consisting of disjunctions only can be instantiated using an anonymous IBE scheme (such as those of [19, 2, 3]) and still achieve the same level of privacy as our scheme provides. Briefly, a key for the predicate $(X = a \text{ OR } X = b)$ consists of the two IBE keys for a and b, and one decrypts by trying all of the keys in one's possession. The anonymity property of the IBE guarantees that the scheme is weakly attribute hiding, but the scheme clearly does not satisfy the strong attribute hiding property. To achieve the stronger form of privacy for disjunctions — which was one of the principal achievements of [28] — it appears that new techniques are needed.

We note that the construction of disjunction predicates using anonymous IBE gives constant-size ciphertexts with key sizes linear in the number of terms in the disjunction, while our instantiation using inner products gives constant-size keys and linear-size ciphertexts.

6 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. 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, 8, 33] can be adapted to lattice based constructions.

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, 18, 17] could be used to help us move towards this goal.

Acknowledgments. The authors thank Dan Boneh, Brent Waters and Hoeteck Wee for helpful discussions.

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions." *J. Cryptology* **21** (2008), 350–391.

[2] S. Agrawal, D. Boneh, and X. Boyen. "Efficient lattice (H)IBE in the standard model." In *Advances in Cryptology — EUROCRYPT 2010*, ed. H. Gilbert, Springer LNCS **6110** (2010), 553–572. Full version at http://crypto.stanford.edu/~dabo/pubs/papers/latticebb.pdf.

[3] S. Agrawal, D. Boneh, and X. Boyen. "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE." In *Advances in Cryptology — CRYPTO '10*, ed. T. Rabin, Springer LNCS **6223** (2010), 98–115.

[4] S. Agrawal and X. Boyen. "Identity-based encryption from lattices in the standard model." Manuscript (2009). Available at http://www.cs.stanford.edu/~xb/ab09/.

[5] M. Ajtai. "Generating hard instances of the short basis problem." In *Automata, Languages, and Programming — ICALP '99*, ed. J. Wiedermann, P. van Emde Boas, and M. Nielsen, Springer LNCS **1644** (1999), 1–9.

[6] J. Alwen and C. Peikert. "Generating shorter bases for hard random lattices." In *STACS* (2009), 75–86. Full version available at `http://www.cc.gatech.edu/~cpeikert/pubs/shorter.pdf`.

[7] N. Attrapadung and H. Imai. "Conjunctive broadcast and attribute-based encryption." In *Pairing-Based Cryptography — Pairing '09*, ed. H. Shacham and B. Waters, Springer LNCS **5671** (2009), 248–265.

[8] 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 — PKC '10*, ed. P. Q. Nguyen and D. Pointcheval, Springer LNCS **6056** (2010), 384–402.

[9] J. Bethencourt, A. Sahai, and B. Waters. "Ciphertext-policy attribute-based encryption." In *IEEE Symposium on Security and Privacy* (2007), 321–334.

[10] C. Blundo, V. Iovino, and G. Persiano. "Predicate encryption with partial public keys." In *Cryptology and Network Security — CANS 2010*, ed. S.-H. Heng, R. N. Wright, and B.-M. Goi, Springer LNCS **6467** (2010), 298–313.

[11] D. Boneh and X. Boyen. "Efficient selective-ID secure identity-based encryption without random oracles." In *Advances in Cryptology — EUROCRYPT '04*, ed. C. Cachin and J. Camenisch, Springer LNCS **3027** (2004), 223–238.

[12] D. Boneh and X. Boyen. "Secure identity based encryption without random oracles." In *Advances in Cryptology — CRYPTO '04*, ed. M. Franklin, Springer LNCS **3152** (2004), 443–459.

[13] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. "Public key encryption with keyword search." In *Advances in Cryptology — EUROCRYPT '04*, ed. C. Cachin and J. Camenisch, Springer LNCS **3027** (2004), 506–522.

[14] D. Boneh and M. Franklin. "Identity-based encryption from the Weil pairing." *SIAM J. Comput.* **32** (2003), 586–615. Extended abstract in *CRYPTO '01*.

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

[16] D. Boneh and B. Waters. "Conjunctive, subset, and range queries on encrypted data." In *Theory of Cryptography — TCC '07*, ed. S. Vadhan, Springer LNCS **4392** (2007), 535–554.

[17] Z. Brakerski and V. Vaikuntanathan. "Efficient fully homomorphic encryption from (standard) LWE." In submission (2011).

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

[19] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. "Bonsai trees, or, how to delegate a lattice basis." In *Advances in Cryptology — EUROCRYPT '10*, ed. H. Gilbert, Springer LNCS **6110** (2010), 523–552.

[20] M. Chase. "Multi-authority attribute based encryption." In *Theory of Cryptography — TCC '07*, ed. S. Vadhan, Springer LNCS **4392** (2007), 515–534.

[21] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data." *SIAM J. Comput.* (2008), 97–139.

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

[23] C. Gentry. *A fully homomorphic encryption scheme*. Ph.D. dissertation, Stanford University (2009). Available at http://crypto.stanford.edu/craig.

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

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

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

[27] 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* (2006), 89–98.

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

[29] 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 *Advances in Cryptology — EUROCRYPT '10*, ed. H. Gilbert, Springer LNCS **6110** (2010), 62–91.

[30] A. Litvak, A. Pajor, M. Rudelson, and N. Tomczak-Jaegermann. "Smallest singular value of random matrices and geometry of random polytopes." *Advances in Mathematics* **195** (2005), 491–523.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

A Probability

Let X and Y be two random variables taking values in some countable set Ω. We define the *statistical distance*, denoted $\Delta(X;Y)$, to be

$$\Delta(X;Y) := \frac{1}{2}\sum_{s\in\Omega}\big|\Pr[X=s]-\Pr[Y=s]\big|$$

If $X(n)$ and $Y(n)$ are ensembles of random variables, we say X and Y are *statistically close* or *statistically indistinguishable* if $\Delta(X;Y)$ is a negligible function of n.

The norm of a random matrix. Let S^m denote the m-sphere; i.e., the set of all vectors in $\mathbb{R}^{m+1}$ of length 1. We define the norm of a matrix $\mathbf{R}\in\mathbb{R}^{k\times m}$ to be $\sup_{\mathbf{x}\in S^{m-1}}\|\mathbf{R}\mathbf{x}\|$. Then we have the following:

Lemma A.1 ([30, Fact 2.4],[2, Lemma 15]). *Let* $\mathbf{R}$ *be a* $k\times m$ *matrix chosen at random from* $\{-1,1\}^{k\times m}$. *Then* $\Pr[\|\mathbf{R}\| > 12\sqrt{k+m}] < e^{-(k+m)}$.

Randomness extraction. We will use the following lemma, which follows from a generalization of the leftover hash lemma due to Dodis et al. [21]. Agrawal, Boneh, and Boyen [2] prove the lemma for prime moduli q; we observe that the result extends to square-free values of q by the Chinese remainder theorem.

Lemma A.2 ([2, Lemma 13]). *Suppose that $m > (n+1)\lg q + \omega(\log n)$ and that $q > 2$ is square free. Let $\mathbf{R}$ be an $m \times k$ matrix chosen uniformly in $\{1, -1\}^{m \times k} \bmod q$ where $k = k(n)$ is polynomial in n. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}^{n \times m}$ and $\mathbb{Z}^{n \times k}$ respectively. Then for all vectors $\mathbf{w} \in \mathbb{Z}^m$, the distribution $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^t\mathbf{w})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^t\mathbf{w})$.*