Advanced cryptography from lattice isomorphism—new constructions of IBE and FHE

Huck Bennett* Zhengnan Lai† Noah Stephens-Davidowitz‡

July 28, 2026

Abstract

We show how to translate some of the more advanced techniques used in LWE-based cryptography to the setting of lattice-isomorphism-based cryptography, which was recently introduced by Ducas and van Woerden [Eurocrypt, 2022]. In particular, we show constructions of two powerful cryptographic primitives, identity-based encryption (IBE) and leveled fully homomorphic encryption (FHE). We then prove their security under the assumption that a suitable version of the Lattice Isomorphism Problem is hard.

Our constructions use quite general and modular techniques, and we expect them to have other applications. Specifically, we show that the now-ubiquitous techniques introduced by Gentry, Peikert, and Vaikuntanathan [STOC, 2008] to build IBE and Gentry, Sahai, and Waters [Crypto, 2013] to build FHE can be made to work in our setting using any sufficiently "nice" lattice.

*University of Colorado Boulder. huck.bennett@colorado.edu. Supported in part by NSF Grant No. CCF-2432132.

†Cornell University. zl345@cornell.edu.

‡Cornell University. noahsd@gmail.com. This work was supported by NSF Grant No. CCF-2312296, a Packard Foundation Fellowship, and a generous gift from Google.

Contents

1 Introduction **1**
- 1.1 Our results . . . 3
- 1.2 Related work . . . 7

2 Preliminaries **7**
- 2.1 Lattice basics . . . 8
- 2.2 Gaussians and discrete Gaussians . . . 8
- 2.3 Smooth sublattices . . . 11
- 2.4 From lattices to quadratic forms . . . 12
- 2.5 The Lattice Isomorphism Problem . . . 15
- 2.6 Identity-based encryption . . . 17
- 2.7 Leveled fully homomorphic encryption . . . 18

3 A public-key encryption scheme that we will use to build IBE **19**
- 3.1 Correctness . . . 20
- 3.2 Security . . . 20

4 Identity-based encryption **21**
- 4.1 Correctness . . . 21
- 4.2 Full security . . . 22
- 4.3 Putting everything together . . . 24
- 4.4 Instantiating our IBE scheme with specific quadratic forms and parameters . . . 24

5 A public-key encryption scheme that we will use to build FHE **25**
- 5.1 The scheme . . . 27
- 5.2 Correctness . . . 28
- 5.3 Security . . . 29

6 Fully homomorphic encryption **30**
- 6.1 The encryption scheme . . . 30
- 6.2 Security . . . 31
- 6.3 The Eval function and homomorphic operations . . . 31
- 6.4 Putting everything together . . . 34
- 6.5 Instantiating our FHE scheme with specific quadratic forms and parameters . . . 34
- 6.6 A note on bootstrapping . . . 36

1 Introduction

A lattice $\mathcal{L} \subset \mathbb{R}^n$ is the set of all integer linear combinations of n linearly independent basis vectors $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{R}^{n \times n}$, i.e.,

$$\mathcal{L} := \mathcal{L}(\boldsymbol{B}) = \{z_1 \boldsymbol{b}_1 + \cdots + z_n \boldsymbol{b}_n \ : \ z_1, \ldots, z_n \in \mathbb{Z}\} \ .$$

Lattices play an extremely important role in cryptography. Indeed, lattice-based cryptography is nearing widespread deployment. In particular, because of their apparent security against quantum adversaries, lattice-based constructions were recently standardized by NIST [Nat24a, Nat24b] as the preferred way to implement both public-key encryption [ABD+21] (more formally, key encapsulation) and digital signatures [BDK+21, FHK+20].

Perhaps even more excitingly, lattices can also be used to construct much more powerful cryptographic primitives like fully homomorphic encryption (FHE) [Gen09, BV11, BGV12, GSW13, BV14, AP14, DM15], identity-based encryption (IBE) [GPV08, AB09, ABB10, CHKP10], fully homomorphic signatures [BF11, GVW15], attribute-based encryption (ABE) [GVW13, BGG+14, Tsa19], functional encryption [AFV11, ADCM12, Xag13, GKP+13, ALS16], and many more. (See [Pei16] for a survey.) All of these schemes are secure under the assumption that certain computational problems associated with the geometry of a lattice—like the problem of estimating the length of the shortest non-zero vector in a lattice—are hard (in many instances, even in the worst case [Ajt96, Reg05, Pei09]).

In fact, essentially all of these schemes have their security based on variants of a *single* computational hardness assumption, known as the Learning With Errors (LWE) assumption [Reg05]. (To the authors' knowledge, the only exceptions in the many schemes referenced above are [Gen09, BF11] and [FHK+20], which are respectively based on an ad-hoc assumption and on NTRU [HPS98].) Evidently, LWE is a remarkably useful and versatile tool. So, it is natural to ask whether there are also other ways of using lattices to build cryptography.

Recently, Ducas and van Woerden [DvW22] and Bennett, Ganju, Peetathawatchai, and Stephens-Davidowitz [BGPS23] proposed a different way to do lattice-based cryptography based on the *Lattice Isomorphism Problem* (LIP). At the heart of the constructions in [DvW22, BGPS23], there is a fixed underlying "nice" lattice $\mathcal{L}_{\mathsf{nice}}$ like $\mathbb{Z}^n$ or the Barnes-Wall lattice that is known to all.[1] $\mathcal{L}_{\mathsf{nice}}$ is "nice" in the sense that some computational problem—like the problem of determining whether a given point $\boldsymbol{c} \in \mathbb{R}^n$ is close or far from the lattice—is easy when restricted to $\mathcal{L}_{\mathsf{nice}}$ but hard for typical lattices $\mathcal{L}$.

To generate the public and secret keys for these cryptosystems, one can imagine sampling an orthogonal transformation $\boldsymbol{O} \in \mathsf{O}_n(\mathbb{R})$ and taking the secret key to be the corresponding transformation $\boldsymbol{B}_{\mathsf{sk}} := \boldsymbol{O}\boldsymbol{B}_{\mathsf{nice}}$ of some fixed, known "nice" basis $\boldsymbol{B}_{\mathsf{nice}}$ of $\mathcal{L}_{\mathsf{nice}}$. The public key can be thought of as a "bad" basis $\boldsymbol{B}_{\mathsf{pk}}$ of $\boldsymbol{O}\mathcal{L}_{\mathsf{nice}} = \mathcal{L}(\boldsymbol{B}_{\mathsf{sk}})$.[2] This is useful because bases are

[1] Ducas and van Woerden considered general "nice" lattices $\mathcal{L}_{\mathsf{nice}}$, which they referred to as "remarkable." Bennett, Ganju, Peetathawatchai, and Stephens-Davidowitz restricted their attention to $\mathcal{L}_{\mathsf{nice}} = \mathbb{Z}^n$.

[2] In both [DvW22, BGPS23], one does not directly work with bases $\boldsymbol{B}$ of an orthogonal transformation $\boldsymbol{O}\mathcal{L}_{\mathsf{nice}}$ of the nice lattice. Instead, it is far more mathematically elegant to work with the associated quadratic forms $\boldsymbol{Q} := \boldsymbol{B}^\top \boldsymbol{B}$. This effectively quotients out by orthogonal transformations (i.e., $(\boldsymbol{O}\boldsymbol{B})^\top \boldsymbol{O}\boldsymbol{B} = \boldsymbol{B}^\top \boldsymbol{B}$ for any $\boldsymbol{O} \in \mathsf{O}_n(\mathbb{R})$) while preserving the Euclidean geometry. So, by working with $\boldsymbol{Q}$ instead of $\boldsymbol{B}$, one effectively works directly with the equivalence class of all orthogonal transformations of a fixed lattice, rather than the lattice itself. This allows us to ignore orthogonal transformations and some thorny computational issues that would otherwise arise. In the sequel, we also work with quadratic forms. (See Section 2.4.) But, in the introduction, we work with bases of lattices because we find them to be more intuitive.

certainly not unique. For example, one could take $\boldsymbol{B}_{\mathsf{nice}} := \boldsymbol{I}_2$ to be the nice basis of $\mathcal{L}_{\mathsf{nice}} = \mathbb{Z}^2$, and $\boldsymbol{B}_{\mathsf{pk}} := \boldsymbol{O}\boldsymbol{B}_{\mathsf{bad}}$ where[3]

$$\boldsymbol{B}_{\mathsf{bad}} := \begin{pmatrix} 12345 & 1234 \\ 2471 & 247 \end{pmatrix}.$$

(Interestingly, this paradigm of "nice bases and bad bases" is often used as a simple way to give some intuition for how lattice-based cryptography works—e.g., in a talk to a general computer science audience. However, most LWE-based constructions do not actually follow this paradigm in any real sense.)

To prove security, we assume that it is hard to distinguish a "bad" basis $\boldsymbol{B}_{\mathsf{pk}}$ of $\mathcal{L} = \boldsymbol{O}\mathcal{L}_{\mathsf{nice}}$ from a basis $\boldsymbol{B}'_{\mathsf{pk}}$ of an orthogonal transformation $\boldsymbol{O}'$ of some *other* lattice $\mathcal{L}'$, and we carefully choose $\mathcal{L}'$ so that breaking our cryptographic scheme is simply information-theoretically impossible when $\mathcal{L}_{\mathsf{nice}}$ is replaced by $\mathcal{L}'$. E.g., for an encryption scheme, we choose $\mathcal{L}'$ so that the resulting distribution of ciphertexts $\boldsymbol{c}'_0$ and $\boldsymbol{c}'_1$ encrypting 0 and 1 respectively are statistically close, and we immediately see that the scheme is secure as long as it is hard to determine whether the input basis is a basis for an orthogonal transformation of $\mathcal{L}_{\mathsf{nice}}$ or of $\mathcal{L}'$.[4]

Using this approach, Ducas and van Woerden [DvW22] proposed a public-key encryption scheme (formally, a KEM), a signature scheme, and a zero-knowledge proof of knowledge and showed security under the assumption that it is hard to distinguish a basis of (an orthogonal transformation of) a sufficiently "nice" lattice from a very different lattice. Bennett, Ganju, Peetathawatchai, and Stephens-Davidowitz [BGPS23] proposed a different public-key encryption scheme using "good and bad bases" of $\mathbb{Z}^n$ and similarly showed its security under the assumption that it is hard to distinguish a basis of (an orthogonal transformation of) $\mathbb{Z}^n$ from a basis of a lattice with "very different geometry than $\mathbb{Z}^n$." Both works showed worst-case to average-case reductions, effectively showing a way to sample the public key basis $\boldsymbol{B}_{\mathsf{pk}}$ so that the average-case problem of distinguishing $\boldsymbol{B}_{\mathsf{pk}}$ from $\boldsymbol{B}'_{\mathsf{pk}}$ is provably as hard as the worst-case problem of distinguishing *any* basis of any orthogonal transformation of $\mathcal{L}_{\mathsf{nice}}$ from any basis of any orthogonal transformation of $\mathcal{L}'$.

This is now called *lattice-isomorphism-based* cryptography. To see where the name comes from, first note that two lattices $\mathcal{L}_0, \mathcal{L}_1$ are said to be *isomorphic* if $\mathcal{L}_0 = \boldsymbol{O}\mathcal{L}_1$ for some orthogonal linear transformation $\boldsymbol{O} \in \mathsf{O}_n(\mathbb{R})$. The problem of determining whether two bases generate isomorphic lattices is naturally known as the *Lattice Isomorphism Problem (*LIP*)*. Thus, the name "lattice-isomorphism-based cryptography."[5]

[3]Notice that we do in fact have $\mathcal{L}(\boldsymbol{B}_{\mathsf{bad}}) = \mathcal{L}(\boldsymbol{B}_{\mathsf{nice}}) = \mathbb{Z}^2$, which can be seen by noting that

$$\boldsymbol{B}_{\mathsf{bad}} \cdot \begin{pmatrix} 247 & -1234 \\ -2471 & 12345 \end{pmatrix} = \boldsymbol{I}_2 .$$

Therefore $\boldsymbol{B}_{\mathsf{pk}} := \boldsymbol{O}\boldsymbol{B}_{\mathsf{bad}}$ similarly generates the same lattice as $\boldsymbol{B}_{\mathsf{sk}} = \boldsymbol{O}\boldsymbol{B}_{\mathsf{nice}}$. Of course, this is just for illustrative purposes and one should not actually use such schemes in two dimensions.

[4]One could imagine simply directly assuming that the underlying computational problem of breaking the scheme is hard, rather than factoring through $\mathcal{L}'$. E.g., for an encryption scheme, one might directly assume that $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$ are computationally indistinguishable. But, it seems more sound to base security on a less contrived assumption.

[5]Really, this is cryptography based on a particular promise problem variant of LIP in which in the YES case $\boldsymbol{B}$ generates an orthogonal transformation of some fixed very special lattice $\mathcal{L}_{\mathsf{nice}}$ and in the NO case $\boldsymbol{B}$ generates an orthogonal transformation of some other fixed lattice $\mathcal{L}'$ that "has very different geometry from $\mathcal{L}_{\mathsf{nice}}$." It seems that this promise problem is quite a bit easier than LIP in general although it is still thought to be hard. So the connection between such cryptographic schemes and LIP is a bit subtle. Nevertheless, we follow the literature in calling this "lattice-isomorphism-based cryptography."

In the few years since [DvW22, BGPS23], there has been quite a remarkable amount of excitement around LIP and lattice-isomorphism-based cryptography. In particular, there has been a flurry of work studying LIP itself and related problems [DG23, Duc24, BN24, MPPW24, BBCK24, BN25, APvW25, LJPW25, SLL25, van25, BW25, Mur25, LLM25, CME+25, JWL+25b, BBB+26, Gya26] and another flurry of work building additional lattice-isomorphism-based cryptography [dCBZI+24, ARW24, BMM25, JWL+25a, KLM+25, LRvW25], including a signature scheme that has been proposed for standardization [DPPvW23, BBD+25]. Indeed, there seems to be quite a large design space to explore here.[6]

Of course, lattice-isomorphism-based cryptography is still nowhere near as mature as traditional lattice-based cryptography based on LWE, and it is so far unclear how such constructions compare to traditional lattice-based constructions.

A natural next step in the development of lattice-isomorphism-based cryptography is to ask whether there are constructions of more advanced cryptographic primitives in this framework. In other words, can we use the ideas above to build the powerful cryptographic primitives—like FHE, IBE, ABE, and Functional Encryption—that we already know how to build from LWE? Such schemes are inherently interesting, and they might lead to better security, improved efficiency, or even constructions of novel cryptographic primitives.

Perhaps the most natural advanced cryptographic primitive to try to build first in this context is FHE, which is a very important cryptographic primitive that now has many different lattice-based constructions but no non-lattice-based constructions. Indeed, just recently, Branco, Malavolta, and Maradni [BMM25] and Leporati, Rovida, and van Woerden [LRvW25] proposed constructions of variants of lattice-isomorphism-based FHE. Specifically, [BMM25] constructed an encryption scheme capable of homomorphically evaluating branching programs.[7] [LRvW25] outlined two whole frameworks for building FHE schemes that capture many known constructions, and which they say can likely be instantiated from suitable versions of lattice isomorphism. But, they do not formally prove security and correctness of the resulting lattice-isomorphism-based scheme. ([LRvW25] were more interested in their frameworks than in particular instantiations. They do prove security formally for non-lattice-isomorphism-based instantiations of their framework, and they give an informal argument for security of their lattice-isomorphism-based instantiations.)

1.1 Our results

We continue the study of constructions of advanced lattice-isomorphism-based cryptographic primitives. In particular, we demonstrate that some of the core techniques used to build advanced primitives from LWE can be modified to construct such primitives from lattice isomorphism as well.

Our concrete contributions are constructions of two advanced cryptographic primitives—IBE and leveled FHE. As we will discuss, our constructions closely parallel two celebrated constructions from traditional lattice-based cryptography, the Gentry, Peikert, and Vaikuntanathan IBE scheme [GPV08], and the Gentry, Sahai, and Waters FHE scheme [GSW13].

[6] In the original work of [DvW22], they observed that constructions of truly "remarkable" lattices could potentially yield better security and efficiency or even novel functionalities. That particular direction is still largely unexplored, and it is still quite exciting.

[7] [BMM25] says that the generic technique of bootstrapping can be used to make this scheme work for arbitrary circuits, therefore achieving true FHE. This seems plausible, but the details are not worked out in [BMM25]. See Section 6.6 for a brief discussion of the subtleties involved in applying bootstrapping in this context.

Identity-based encryption. First, we show (in Sections 3 and 4) the first construction of lattice-isomorphism-based identity-based encryption (IBE). Recall that IBE is a variant of public-key encryption in which "Alice and Bob's public keys are simply their names." More specifically, in an IBE scheme, a trusted authority can use a master secret key msk to generate a secret key $\mathsf{sk}_{\mathsf{id}}$ corresponding to any identity $\mathsf{id} \in \{0,1\}^*$, e.g., the names "Alice" and "Bob" (in ASCII, say). Anyone can then use a master public key mpk together with the associated id to encrypt a message μ so that it can be decrypted using just $\mathsf{sk}_{\mathsf{id}}$. (I.e., $\mathsf{Dec}(\mathsf{sk}_{\text{Alice}}, \mathsf{Enc}(\mathsf{mpk}, \text{"Alice"}, \mu)) = \mu$.) Security requires that it is difficult for an adversary to distinguish an encryption under some identity id of 0 from an encryption under id of 1, even if the adversary can request many secret keys for many identities $\mathsf{id}' \neq \mathsf{id}$.

We construct an IBE scheme in the style of the celebrated LWE-based IBE scheme of Gentry, Peikert, and Vaikuntanathan [GPV08]. We then prove the security of our scheme assuming the hardness of a suitable version of LIP (in the random oracle model), which appears to be incomparable to the version of LWE used in [GPV08].

The scheme itself is quite simple and follows the paradigm of "good and bad bases" described above. In particular, mpk is (effectively) a bad basis $\boldsymbol{B}_{\mathsf{mpk}}$ for an orthogonal transformation of some "nice" lattice $\mathcal{L}_{\mathsf{nice}}$ and the master secret key msk is effectively a good basis of the same lattice. Given an identity id, the trusted authority uses its good basis to find a (suitably sampled) dual lattice vector $\boldsymbol{w}$ that is close to $\boldsymbol{t} := \mathcal{H}(\mathsf{id})$ (where $\mathcal{H}$ is a random oracle with an appropriate range) and sets $\mathsf{sk}_{\mathsf{id}} := \boldsymbol{w}$.

An encryption of a bit $\mu \in \{0,1\}$ then has the form $(\boldsymbol{c} := \boldsymbol{y} + \boldsymbol{e}, d := \langle \boldsymbol{y}, \boldsymbol{t} \rangle + \mu/2 \bmod 1)$, where $\boldsymbol{y}$ is a lattice vector and $\boldsymbol{e} \in \mathbb{R}^n$ is "short". Since $\boldsymbol{y}$ is a lattice vector and $\boldsymbol{w}$ is a dual lattice vector, their inner product is an integer. Therefore, when reducing modulo 1, we can ignore the inner product of $\boldsymbol{w}$ and $\boldsymbol{y}$ to see that

$$\langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{w} \rangle = \langle \boldsymbol{e}, \boldsymbol{t} - \boldsymbol{w} \rangle + \langle \boldsymbol{y}, \boldsymbol{t} \rangle = \langle \boldsymbol{e}, \boldsymbol{t} - \boldsymbol{w} \rangle + d - \mu/2 \bmod 1 \ .$$

One can then decrypt by computing $\langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{w} \rangle - d$ and checking whether the result is close to an integer. As long as $\langle \boldsymbol{e}, \boldsymbol{t} - \boldsymbol{w} \rangle$ is much smaller than 1, this will work. And, since we have chosen $\boldsymbol{e}$ to be short and $\boldsymbol{w}$ to be close to $\boldsymbol{t}$, we can choose our parameters so that this inner product will in fact be small.

For security, we assume that the lattice $\mathcal{L} = \boldsymbol{O}\mathcal{L}_{\mathsf{nice}}$ generated by $\boldsymbol{B}_{\mathsf{mpk}}$ is indistinguishable from some other lattice $\mathcal{L}'$ whose geometry is such that the element d is nearly uniformly random modulo one, even conditioned on $\boldsymbol{c}$. (One must also argue that security holds even against an adversary with access to many secret keys $\mathsf{sk}_{\mathsf{id}}$, but we ignore this here. In Section 4, we prove full security.) At a technical level, we show (using a slightly modified version of an argument from [DvW22]) that it suffices if $\mathcal{L}'$ contains a sufficiently dense sublattice (specifically, a sublattice of sufficiently large rank that has small smoothing parameter).

This scheme will be familiar to anyone who is well-versed in the GPV IBE construction. Indeed, we view our construction (including the details that we have glossed over here for simplicity, but which are included in Sections 3 and 4) as showing that the celebrated techniques of [GPV08] work in the setting of LIP.

In more detail, like in [GPV08], one can view our IBE scheme as a sort of "merger of a signature scheme and a compatible public-key encryption scheme." (Note that this provides the intuition for our scheme, but we do not attempt to make this "merger" formal.) In [GPV08], they introduce the GPV signature scheme and the dual Regev public-key encryption scheme and show that the secret

key of the encryption scheme is essentially a signature of a random string that is included in the public key. They then construct an IBE scheme by taking msk and mpk to be the corresponding keys of the signature scheme and set $\mathsf{sk}_{\mathsf{id}}$ to be a signature of a hash of id (with the hash function modeled as a random oracle).

Our construction also has this form. In particular, the signature scheme underlying our construction (which we do not describe explicitly) is precisely the elegant signature scheme from [DvW22]. We present in Section 3 a public-key encryption scheme, which is a simple variant of the encryption scheme in [DvW22]. This variant of [DvW22]'s scheme is naturally designed to have the property that the secret key can be viewed as a signature of a random string. At a technical level, our scheme differs from [DvW22] in that we replace the strong extractor used by [DvW22] with a simple inner product. We show directly using Fourier-analytic techniques that the inner product is sufficient without changing domains. See Theorem 2.15. (The authors of [BMM25] do something similar, but they use an inner product over a smaller domain because they treat the inner product as a generic extractor. We view Theorem 2.15 as an additional important contribution of the present work, though it is rather technical.)

Our presentation is quite modular in the sense that correctness of our IBE scheme follows more-or-less immediately from correctness of the much simpler public-key encryption scheme that we present in Section 3. Security of our IBE scheme follows from a simulation argument (similar to the proof of security of the signature in [DvW22], which itself follows [GPV08]) together with the security of the public-key encryption scheme.

Fully homomorphic encryption. Second, we show (in Sections 5 and 6) a construction of lattice-isomorphism-based fully homomorphic encryption (FHE). More formally, we obtain a leveled FHE scheme, which is a scheme that allows for homomorphic evaluation of circuits up to a certain depth. We can specifically handle circuits up to any depth that is bounded by some polynomial in the security parameter, matching state-of-the-art constructions from LWE.[8]

Here, we closely follow the elegant LWE-based construction of (leveled) FHE due to Gentry, Sahai, and Waters [GSW13]. At a high level, our scheme works by replacing the Regev public-key encryption scheme on which the GSW FHE scheme is based with a suitable, general variant of the public-key encryption scheme from [BGPS23]. We present this public-key encryption scheme in Section 5, and then show how to make it fully homomorphic in Section 6.

We include detailed discussion of the design decisions behind our public-key encryption scheme and FHE scheme at the start of Section 5. Here, we simply note that our FHE scheme makes heavy use of Micciancio and Peikert's celebrated *gadget matrix* $\boldsymbol{G}$ [MP12] (see Equation (3)). The gadget matrix is a key tool in [GSW13] specifically, but it is also used in *many* beautiful lattice-based cryptographic constructions, and we show how to make it work in the context of lattice isomorphism.

We show how to make gadget-based techniques work relatively painlessly, although we do encounter a subtlety that requires care to get around. The specific subtlety is rather technical and discussed in Section 5, but it amounts to the question of whether $\boldsymbol{s}^\top \boldsymbol{G} \bmod q$ is "large or small," where $\boldsymbol{s}$ is the secret key. If this vector is small, then all valid ciphertexts will decrypt to 0, and the scheme will certainly not be correct. In LWE-based constructions, it is trivial to see that

[8] All known true FHE constructions require additional circular security assumptions, which might also work in our setting. See Section 6.6 for more discussion.

$\boldsymbol{s}^\top \boldsymbol{G} \bmod q$ is large,[9] but there seems to be no guarantee that this remains true here.[10] We solve this by replacing $\boldsymbol{G}$ with $\boldsymbol{H} := (\boldsymbol{h}, \boldsymbol{G})$ where $\boldsymbol{h}$ is chosen so that $\langle \boldsymbol{s}, \boldsymbol{h} \rangle \bmod q$ is large. However, if we chose $\boldsymbol{h}$ in such a way that this depended on the secret key $\boldsymbol{s}$, this could break security (and certainly we would not be able to prove security). So, we instead choose $\boldsymbol{h}$ randomly from a suitable set and show that with high probability this inner product will be large. (To make this work with high probability, we actually take our secret key to be a whole matrix $\boldsymbol{S}$ rather than a single vector $\boldsymbol{s}$. See Section 5 for a detailed discussion.)

We briefly compare the security assumption that we need for our LIP-based FHE scheme to the assumptions needed by LWE-based schemes. We prove that our FHE scheme is secure assuming the hardness of distinguishing orthogonal linear transformations of lattices $\mathcal{L}_{\mathsf{nice}}$ and $\mathcal{L}'$ satisfying certain properties. For the concrete choices of $\mathcal{L}_{\mathsf{nice}}$ and $\mathcal{L}'$ that we discuss in Section 6.5, this in turn requires the hardness of γ-GapSVP with $\gamma = n^{L+\Theta(1)}$, where L is the depth of the circuit to be homomorphically evaluated and γ-GapSVP is the problem of approximating the length of the shortest non-zero vector in a lattice to within a factor of γ.

Using simple constructions, one can prove security of LWE-based FHE assuming the hardness of γ-GapSVP for a similar factor $\gamma = n^{L+\Theta(1)}$. E.g., one can prove security of the GSW FHE scheme [GSW13] assuming the hardness of LWE with certain parameters, and one can then apply Regev's worst-case to average-case reduction for LWE to γ-GapSVP with $\gamma = n^{L+\Theta(1)}$ to LWE with these parameters. However, [BV14] showed how to use sophisticated techniques to reduce this approximation factor to $\gamma = n^{\Theta(1)}$, independent of L—a weaker assumption than the one that we require (for L larger than some small constant). So, if one compares our scheme to similarly simple LWE-based constructions of FHE, one finds that the corresponding security assumptions are incomparable (or at least not known to be comparable). If one compares to the state of the art, one sees that our assumption is stronger than the relevant version of LWE. (This was originally observed by [BMM25].) We leave it as an open problem to make ideas like those in [BV14] work in our setting in order to achieve security under a weaker hardness assumption. This is closely related to the discussion of bootstrapping in Section 6.6. See also Footnote 24 and Section 6.5 for additional discussion.

Generality and modularity. In our presentation, we endeavor to be general. In particular, we try to use quite natural and quite weak properties of the lattice $\mathcal{L}_{\mathsf{nice}}$ in order to prove correctness of our schemes, and we try to use similarly weak and natural properties of the lattice $\mathcal{L}'$ in order to prove security. We then plug in simple but careful choices of $\mathcal{L}_{\mathsf{nice}}$ and $\mathcal{L}'$ that were originally proposed by Ducas and van Woerden [DvW22]. In particular, in both schemes, we take $\mathcal{L}_{\mathsf{nice}} = a\mathbb{Z}^{n/2} \oplus b\mathbb{Z}^{n/2}$ and $\mathcal{L}' = a'\mathbb{Z}^{n/2} \oplus b'\mathbb{Z}^{n/2}$ for appropriate a, b, a', and b'. But, we work to avoid using any unnecessary properties of these lattices—such as the fact that decoding is easy. We note that this modularity allows for drop-in replacement of other "nice" lattices, and we view this as a strength of our work and, more generally, of lattice-isomorphism-based cryptography. It is an interesting open question to find new useful examples of "nice" families of lattices that are suitable for our constructions (and other similar constructions).

[9] Here, we are parameterizing the scheme so that the last coordinate of $\boldsymbol{s}$ is always 1, in which case this fact is guaranteed by the fact that the last column of the gadget matrix is $(q/2)\boldsymbol{e}_n$. This parameterization is often not made explicit.

[10] In their informal construction of FHE from LIP (which they present as just one part of a more general framework), [LRvW25] quite reasonably simply assume that $\boldsymbol{s}^\top \boldsymbol{G} \bmod q$ is large. In their notation, this amounts to the assumption that $\langle \boldsymbol{d}_1, \boldsymbol{s} \rangle = 1 \bmod p$.

1.2 Related work

As we noted above, our IBE scheme is really an adaptation to the LIP setting of the beautiful LWE-based IBE scheme of Gentry, Peikert, and Vaikuntanathan [GPV08]. Analogously, our FHE scheme is an adaptation of the elegant LWE-based FHE scheme of Gentry, Sahai, and Waters [GSW13]. Furthermore, many of the specific "lattice-isomorphism-based" methods that we use to construct our schemes are heavily inspired by the ideas in Ducas and van Woerden [DvW22] and in Bennett, Ganju, Peetathawatchai, and Stephens-Davidowitz [BGPS23]. Indeed, our IBE scheme is heavily inspired by both the signature and the public-key encryption scheme in [DvW22]—and can be viewed as a "merger" of the two, as we described above. Similarly, the public-key encryption scheme that we present in Section 5 (which we then use in Section 6 to build FHE) is a variant of the public-key encryption scheme in [BGPS23].

The other works that are most closely related to ours are the two other works that consider lattice-isomorphism-based FHE. Indeed, Branco, Malavolta, and Maradni built a lattice-isomorphism-based encryption scheme that can homomorphically evaluate branching programs [BMM25], while we build an encryption scheme that can handle the much larger class of polynomial-depth circuits.[11] The construction of [BMM25] is quite a bit different than ours. They actually start out with an encryption scheme that is quite similar to the one that we present in Section 3, which we use in our IBE construction. However, they use Micciancio's "ground up" approach to convert this scheme into a homomorphic scheme [Mic19], whereas we start with our scheme from Section 5 and make it homomorphic using the very different approach from [GSW13].

Leporati, Rovida, and van Woerden also considered building lattice-based FHE from assumptions "beyond LWE" [LRvW25]. They suggested two frameworks for building (leveled) FHE, one generalizing [BGV12], and the other generalizing [GSW13]. When they instantiate the [GSW13]-based framework using LIP, they naturally obtain a construction that looks quite similar to our own [GSW13]-based scheme. In particular, they similarly use the gadget matrix $\boldsymbol{G}$. However, they are more interested in their general framework, so their construction is heuristic and not proven. In contrast, we are certainly not nearly as general, but we provide a full proof of correctness and security.

Acknowledgments

The authors thank Chris Peikert for helpful discussion and the anonymous Crypto 2026 reviewers for their useful comments.

2 Preliminaries

We typically use bold capital letters for matrices, like $\boldsymbol{X} \in \mathbb{Z}^{n \times m}$, and lower-case bold letters for vectors, like $\boldsymbol{x} \in \mathbb{R}^n$. All vectors are column vectors except where otherwise noted. For a positive integer q, we write $\mathbb{T}_q := \{0, 1/q, \ldots, (q-1)/q\}$, which we think of as an additive group modulo 1.

We use $\mathcal{S}_n^{>0}(\mathbb{R})$ and $\mathcal{S}_n^{>0}(\mathbb{Z})$ to denote the set of $n \times n$ symmetric positive definite matrices over the reals and the integers, respectively.

[11] [BMM25] claim that generic bootstrapping-based techniques can be used to convert this to a fully homomorphic public-key encryption scheme, but they do not prove correctness or security of the resulting scheme. We briefly discuss some of the subtleties that arise in this context in Section 6.6.

2.1 Lattice basics

A lattice $\mathcal{L} \subset \mathbb{R}^d$ is the set

$$\mathcal{L} := \mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) := \{z_1\boldsymbol{b}_1 + \cdots + z_n\boldsymbol{b}_n \ : \ z_i \in \mathbb{Z}\}$$

of all integer linear combinations of the vectors in a basis $\boldsymbol{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{R}^{d\times n}$. We call n the *rank* of the lattice and d the *ambient dimension*. We often assume that $d = n$, in which case we say that the lattice $\mathcal{L}$ has *full rank*.

Similarly, a sublattice $\mathcal{L}' \subseteq \mathcal{L}$ of $\mathcal{L}$ is called *full rank* if $\mathcal{L}$ and $\mathcal{L}'$ have the same rank. For a full-rank sublattice $\mathcal{L}' \subseteq \mathcal{L}$, we write $\mathcal{L}/\mathcal{L}'$ for the set of cosets of $\mathcal{L}$ modulo $\mathcal{L}'$, which is always a finite set.

Crucially, the basis $\boldsymbol{B}$ of a lattice is not unique. In particular, one can show that $\mathcal{L}(\boldsymbol{B}) = \mathcal{L}(\boldsymbol{B}')$ if and only if $\boldsymbol{B}' = \boldsymbol{B}\boldsymbol{U}$ for a matrix $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$, i.e., an integral matrix $\boldsymbol{U} \in \mathbb{Z}^{n\times n}$ with $\det(\boldsymbol{U}) = \pm 1$. Such matrices $\boldsymbol{U}$ are called *unimodular* matrices. In particular, unimodular matrices are precisely matrices $\boldsymbol{U}$ with the property that $\boldsymbol{U}\mathbb{Z}^n = \mathbb{Z}^n$—i.e., they are precisely the linear bijections on $\mathbb{Z}^n$.

A *sublattice* $\mathcal{L}' \subset \mathcal{L}$ of rank $k \le n$ is *primitive* if there exists a basis $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{R}^{d\times n}$ of $\mathcal{L}$ such that $\mathcal{L}' = \mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$. One can show that a sublattice $\mathcal{L}' \subset \mathcal{L}$ is primitive if and only if $\mathcal{L}' = \mathcal{L} \cap \mathrm{span}(\mathcal{L}')$.

For a target vector $\boldsymbol{t} \in \mathbb{R}^d$ and a lattice $\mathcal{L} \subset \mathbb{R}^d$, we write

$$\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) := \min_{\boldsymbol{y}\in\mathcal{L}} \|\boldsymbol{y} - \boldsymbol{t}\| \ .$$

For a lattice $\mathcal{L} \subset \mathbb{R}^d$, the *dual lattice* is the set

$$\mathcal{L}^* := \{\boldsymbol{w} \in \mathrm{span}(\mathcal{L}) \ : \ \forall \boldsymbol{y} \in \mathcal{L}, \ \langle \boldsymbol{w}, \boldsymbol{y}\rangle \in \mathbb{Z}\}$$

of all vectors in $\mathrm{span}(\mathcal{L})$ that have integer inner product with all vectors in $\mathcal{L}$. If $\mathcal{L} = \mathcal{L}(\boldsymbol{B})$ is generated by some basis $\boldsymbol{B} \in \mathbb{R}^{d\times n}$, then $\mathcal{L}^* = \mathcal{L}(\boldsymbol{B}^*)$ where $\boldsymbol{B}^* := \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}$ is the *dual basis* of $\boldsymbol{B}$. In particular, $\mathcal{L}^*$ is a lattice with the same rank as $\mathcal{L}$, and in the special case when $d = n$ so that $\mathcal{L}$ has full rank, $\boldsymbol{B}^* = \boldsymbol{B}^{-\top}$ is the inverse transpose of $\boldsymbol{B}$.

2.2 Gaussians and discrete Gaussians

The Gaussian mass of a vector $\boldsymbol{y} \in \mathbb{R}^d$ for a parameter $s > 0$ is

$$\rho_s(\boldsymbol{y}) := \exp(-\pi\|\boldsymbol{y}\|^2/s^2).$$

For a lattice $\mathcal{L} \subset \mathbb{R}^d$ and a shift vector $\boldsymbol{t} \in \mathbb{R}^d$, we define the Gaussian mass of the lattice coset $\mathcal{L} - \boldsymbol{t}$ as

$$\rho_s(\mathcal{L} - \boldsymbol{t}) := \sum_{\boldsymbol{y}\in\mathcal{L}} \rho_s(\boldsymbol{y} - \boldsymbol{t}) \ .$$

The *discrete Gaussian distribution* $D_{\mathcal{L},s,\boldsymbol{t}}$ over $\mathcal{L}$ with parameter s centered at $\boldsymbol{t}$ is the probability distribution over $\mathcal{L}$ induced by this measure. That is, for every $\boldsymbol{y} \in \mathcal{L}$,

$$\Pr_{\boldsymbol{X}\sim D_{\mathcal{L},s,\boldsymbol{t}}}[\boldsymbol{X} = \boldsymbol{y}] = \frac{\rho_s(\boldsymbol{y} - \boldsymbol{t})}{\rho_s(\mathcal{L} - \boldsymbol{t})} \ .$$

We write $D_{\mathcal{L},s} := D_{\mathcal{L},s,\boldsymbol{0}}$ for the special case when $\boldsymbol{t} = \boldsymbol{0}$.

For $\varepsilon > 0$, the *smoothing parameter* of a lattice $\mathcal{L}$ is the unique parameter $\eta_\varepsilon(\mathcal{L}) > 0$ such that

$$\rho_{1/\eta_\varepsilon(\mathcal{L})}(\mathcal{L}^*) = 1 + \varepsilon \ .$$

In particular, such a parameter $\eta_\varepsilon(\mathcal{L})$ exists because $\rho_{1/s}(\mathcal{L}^*)$ is a strictly decreasing function of s. Furthermore, for the same reason, notice that $\rho_{1/s}(\mathcal{L}^*) \le 1 + \varepsilon$ if and only if $s \ge \eta_\varepsilon(\mathcal{L})$. (We use this fact quite a bit.)

Notice that the smoothing parameter satisfies that $\eta_\varepsilon(q\mathcal{L}) = q\eta_\varepsilon(\mathcal{L})$ for any positive scalar $q > 0$, and for any *full-rank* sublattice $\mathcal{L}' \subseteq \mathcal{L}$, we have $\eta_\varepsilon(\mathcal{L}) \le \eta_\varepsilon(\mathcal{L}')$.

The smoothing parameter gets its name from the following claim, which shows that when $s \ge \eta_\varepsilon(\mathcal{L})$ for small ε, the function $\boldsymbol{t} \mapsto \rho_s(\mathcal{L} - \boldsymbol{t})$ is nearly constant (i.e., "smooth" like a sheet on a well-made bed).

Claim 2.1. *For any $\varepsilon \in (0, 1)$, any full-rank lattice $\mathcal{L} \subset \mathbb{R}^n$, any $s > 0$, and any shift vector $\boldsymbol{t} \in \mathbb{R}^n$, we have*

$$\rho_s(\mathcal{L} - \boldsymbol{t}) \le \rho_s(\mathcal{L}) \ .$$

If we have $s \ge \eta_\varepsilon(\mathcal{L})$, then

$$\frac{1-\varepsilon}{1+\varepsilon} \cdot \rho_s(\mathcal{L}) \le \rho_s(\mathcal{L} - \boldsymbol{t}) \le \rho_s(\mathcal{L}) \ .$$

We will use the following basic corollary of this claim quite a bit. We include a proof for completeness.

Lemma 2.2. *For any lattice $\mathcal{L} \subset \mathbb{R}^d$, full-rank sublattice $\mathcal{L}' \subseteq \mathcal{L}$, $\varepsilon \in (0, 1/2)$, parameter $s \ge \eta_\varepsilon(\mathcal{L}')$, and shift vector $\boldsymbol{t} \in \mathbb{R}^d$, the distribution $D_{\mathcal{L},s,\boldsymbol{t}} \bmod \mathcal{L}'$ (i.e., the distribution obtained by sampling $\boldsymbol{X} \sim D_{\mathcal{L},s,\boldsymbol{t}}$ and then reducing modulo $\mathcal{L}'$) is within statistical distance 2ε of the uniform distribution over $\mathcal{L}/\mathcal{L}'$.*

Proof. By identifying $\mathrm{span}(\mathcal{L})$ with $\mathbb{R}^n$, we may assume without loss of generality that $\mathcal{L}$ is a full-rank lattice, which implies that $\mathcal{L}'$ is a full-rank lattice as well. For any $\boldsymbol{v} \in \mathcal{L}/\mathcal{L}'$, we can write

$$\Pr_{\boldsymbol{X} \sim D_{\mathcal{L},s,\boldsymbol{t}}}[\boldsymbol{X} = \boldsymbol{v} \bmod \mathcal{L}'] = \frac{1}{\rho_s(\mathcal{L} - \boldsymbol{t})} \cdot \sum_{\boldsymbol{y} \in \mathcal{L}'} \rho_s(\boldsymbol{y} + \boldsymbol{v} - \boldsymbol{t}) = \frac{\rho_s(\mathcal{L}' + \boldsymbol{v} - \boldsymbol{t})}{\rho_s(\mathcal{L} - \boldsymbol{t})} \ .$$

Applying Claim 2.1, we see that

$$\frac{1-\varepsilon}{1+\varepsilon} \cdot \frac{\rho_s(\mathcal{L}')}{\rho_s(\mathcal{L} - \boldsymbol{t})} \le \Pr_{\boldsymbol{X} \sim D_{\mathcal{L},s,\boldsymbol{t}}}[\boldsymbol{X} = \boldsymbol{v} \bmod \mathcal{L}'] \le \frac{\rho_s(\mathcal{L}')}{\rho_s(\mathcal{L} - \boldsymbol{t})} \ .$$

Noting that the probabilities must sum to one, it follows from the upper bound above that

$$1 \le |\mathcal{L}/\mathcal{L}'| \cdot \frac{\rho_s(\mathcal{L}')}{\rho_s(\mathcal{L} - \boldsymbol{t})} \ ,$$

which after plugging in to the lower bound above yields that

$$\Pr_{\boldsymbol{X} \sim D_{\mathcal{L},s,\boldsymbol{t}}}[\boldsymbol{X} = \boldsymbol{v} \bmod \mathcal{L}'] \ge \frac{1-\varepsilon}{1+\varepsilon} \cdot \frac{1}{|\mathcal{L}/\mathcal{L}'|} \ge \frac{1-2\varepsilon}{|\mathcal{L}/\mathcal{L}'|} \ ,$$

and the result follows. □

We will need the following result from [GPV08].

Lemma 2.3 (Slight weakening of [GPV08, Lemma 3.1])**.** *For any lattice $\mathcal{L} \subset \mathbb{R}^d$ with basis $\boldsymbol{B} \in \mathbb{R}^{d \times n}$ and $\varepsilon \in (0, 1/2)$, we have that*

$$\eta_\varepsilon(\mathcal{L}) \leq \|\boldsymbol{B}\| \cdot \sqrt{\log(n/\varepsilon)}.$$

We have the following result due to Banaszczyk [Ban93].

Lemma 2.4. *For any lattice $\mathcal{L} \subset \mathbb{R}^d$ with rank n, any parameter $s > 0$, and any $\alpha \geq 1$,*

$$\rho_{\alpha s}(\mathcal{L}) \leq \alpha^n \rho_s(\mathcal{L}) \ .$$

From Lemma 2.4, we derive the following useful corollary.

Corollary 2.5. *For any lattice $\mathcal{L} \subset \mathbb{R}^d$ with rank n, any prime $q \geq 2$, and any $\boldsymbol{t} \in \mathcal{L}^*/(q\mathcal{L}^*)$ define*

$$\mathcal{L}_{\boldsymbol{t}} := \{\boldsymbol{y} \in \mathcal{L} \ : \ \langle \boldsymbol{t}, \boldsymbol{y} \rangle = 0 \bmod q\} \ .$$

Then, $\mathcal{L}_{\boldsymbol{t}}$ is a full-rank sublattice, and for any $1 \leq \beta \leq q^{1-1/n}$ and any $\varepsilon \in (0, 1)$,

$$\Pr_{\boldsymbol{t} \sim \mathcal{L}^*/(q\mathcal{L}^*)}[\eta_{\varepsilon'}(\mathcal{L}_{\boldsymbol{t}}) \geq \beta q^{1/n} \cdot \eta_\varepsilon(\mathcal{L})] \leq \beta^{-n/2} \ ,$$

where $\varepsilon' := \varepsilon + 2\beta^{-n/2}$.

Proof. It is immediate that $\mathcal{L}_{\boldsymbol{t}}$ is a full-rank sublattice. Notice that

$$(\mathcal{L}_{\boldsymbol{t}})^* = \frac{1}{q} \cdot \{\boldsymbol{w} \in \mathcal{L}^* \ : \ \exists k \in \mathbb{Z}_q, \boldsymbol{w} = k\boldsymbol{t} \bmod q\mathcal{L}^*\} \ .$$

It follows that for any fixed $\boldsymbol{w} \in \mathcal{L}^*/q$ with $\boldsymbol{w} \notin \mathcal{L}^*$,

$$\Pr_{\boldsymbol{t} \sim \mathcal{L}^*/(q\mathcal{L}^*)}[\boldsymbol{w} \in (\mathcal{L}_{\boldsymbol{t}})^*] \leq \frac{1}{q^{n-1}} \ .$$

Therefore, for any $s > 0$, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{t} \sim \mathcal{L}^*/(q\mathcal{L}^*)}[\rho_{1/s}((\mathcal{L}_{\boldsymbol{t}})^*)] &\leq \rho_{1/s}(\mathcal{L}^*) + \frac{1}{q^{n-1}} \cdot \sum_{\boldsymbol{w} \in \mathcal{L}^*/q \setminus \mathcal{L}^*} \rho_{1/s}(\boldsymbol{w}) \\
&= \rho_{1/s}(\mathcal{L}^*) + \frac{1}{q^{n-1}} \rho_{1/s}((\mathcal{L}^*/q) \setminus \mathcal{L}^*) \\
&\leq \rho_{1/s}(\mathcal{L}^*) + \frac{1}{q^{n-1}} \rho_{1/s}(\mathcal{L}^*/q) \ .
\end{aligned}$$

In particular, taking $s := \beta q^{1/n} \eta_\varepsilon(\mathcal{L})$ and setting $\alpha := q^{1-1/n}/\beta \geq 1$, we obtain

$$\rho_{1/s}(\mathcal{L}^*/q) = \rho_{q/s}(\mathcal{L}^*) = \rho_{\alpha/\eta_\varepsilon(\mathcal{L})}(\mathcal{L}^*) \leq \alpha^n \rho_{1/\eta_\varepsilon(\mathcal{L})}(\mathcal{L}^*) = \alpha^n(1+\varepsilon) \leq 2q^{n-1}/\beta^n \ ,$$

where the first inequality is Lemma 2.4. It follows that

$$\mathop{\mathbb{E}}_{\boldsymbol{t} \sim \mathcal{L}^*/(q\mathcal{L}^*)}[\rho_{1/s}((\mathcal{L}_{\boldsymbol{t}})^*) - \rho_{1/s}(\mathcal{L}^*)] \leq 2/\beta^n \ .$$

Noting that $\rho_{1/s}((\mathcal{L}_{\boldsymbol{t}})^*) - \rho_{1/s}(\mathcal{L}^*)$ is non-negative (since $\mathcal{L}^* \subseteq (\mathcal{L}_{\boldsymbol{t}})^*$), we can apply Markov's inequality to see that

$$\Pr_{\boldsymbol{t} \sim \mathcal{L}^*/(q\mathcal{L}^*)}[\rho_{1/s}((\mathcal{L}_{\boldsymbol{t}})^*) - \rho_{1/s}(\mathcal{L}^*) \geq 2/\beta^{n/2}] \leq \frac{1}{\beta^{n/2}} \ .$$

When this event does not occur, we have

$$\rho_{1/s}((\mathcal{L}_{\boldsymbol{t}})^*) \leq \rho_{1/s}(\mathcal{L}^*) + 2/\beta^{n/2} \leq 1 + \varepsilon + 2/\beta^{n/2} \ ,$$

which implies the result, where the last inequality simply uses the fact that $s \geq \beta q^{1/n} \eta_\varepsilon(\mathcal{L}) \geq \eta_\varepsilon(\mathcal{L})$.[12] □

We will need the following simplified version of Banaszczyk's celebrated tail bound on the discrete Gaussian distribution [Ban93].

Theorem 2.6. *For any full-rank lattice $\mathcal{L} \subset \mathbb{R}^n$, shift $\boldsymbol{t} \in \mathbb{R}^n$, and parameter $s \geq \eta_\varepsilon(\mathcal{L})$ for some $\varepsilon \in (0, 1/3)$,*

$$\Pr_{\boldsymbol{x} \sim D_{\mathcal{L},s,\boldsymbol{t}}}[\|\boldsymbol{x} - \boldsymbol{t}\| \geq \sqrt{n}s] < 2^{-n} \ .$$

We will also need the following lemma, due to [MP12]. It shows that the discrete Gaussian distribution is *subgaussian*.

Lemma 2.7 ([MP12, Lemma 2.8])**.** *For lattice $\mathcal{L} \subset \mathbb{R}^d$, parameter $s > 0$, vector $\boldsymbol{v} \in \mathbb{R}^d$, and $r > 0$,*

$$\Pr_{\boldsymbol{y} \sim D_{\mathcal{L},s}}[\langle \boldsymbol{y}, \boldsymbol{v} \rangle \geq r] \leq \exp(-\pi r^2/(s^2\|\boldsymbol{v}\|^2)).$$

Furthermore,

$$\Pr_{\boldsymbol{y} \sim D_{\mathcal{L},s}}[|\langle \boldsymbol{y}, \boldsymbol{v} \rangle| \geq r] \leq 2\exp(-\pi r^2/(s^2\|\boldsymbol{v}\|^2)).$$

2.3 Smooth sublattices

We will also need a slight generalization of the smoothing parameter. Our notation for it is novel (and we hope will prove helpful in future work), though this quantity arises implicitly in a number of works, including [RS17, Dad19, ALS21, DvW22, EWY23]. In particular, our use of it is more-or-less identical to the use in [DvW22]. The idea is that we are sometimes in a situation where $\eta_\varepsilon(\mathcal{L})$ is not small but there is a sublattice $\mathcal{L}' \subset \mathcal{L}$ with small $\eta_\varepsilon(\mathcal{L}')$. Then, the distribution of a Gaussian modulo $\mathcal{L}$ will not be uniform, but it will still have large entropy.

To that end, for a lattice $\mathcal{L} \subset \mathbb{R}^d$ with rank n, an integer $1 \leq k \leq n$, and $\varepsilon > 0$, we define

$$\eta_\varepsilon^{(k)}(\mathcal{L}) := \min_{\mathcal{L}' \subseteq \mathcal{L}} \eta_\varepsilon(\mathcal{L}') \ ,$$

where the minimum is taken over sublattices $\mathcal{L}'$ of rank k. In particular, $\eta_\varepsilon^{(n)}(\mathcal{L}) = \eta_\varepsilon(\mathcal{L})$.

[12] One can get a tighter result here by noting that in fact for such a large s, $\rho_{1/s}(\mathcal{L}^*) \leq 1 + \varepsilon^{\beta^2 q^{2/n}}$, which follows from a convexity argument. (See [CDLP13, Lemma 2.4].) However, in the parameter regime that interests us, the difference is unimportant.

We are interested in this quantity because of the following theorem. Similar arguments appear in [DvW22] and in [BMM25]. In particular, [DvW22] argue that $\boldsymbol{x}$ has entropy and then use a seeded extractor to extract a uniformly random variable from $\boldsymbol{x}$. [BMM25] argue that the extractor can be a linear function, which essentially matches our construction with $\boldsymbol{t}$ being the extractor seed (with some added complications caused by mixing moduli). We instead simply prove directly using lattice-based techniques that $(\boldsymbol{t}, \langle \boldsymbol{x}, \boldsymbol{t} \rangle)$ is close to uniform. (The key difference in our proof is Corollary 2.5.)

Theorem 2.8. *For a lattice $\mathcal{L} \subset \mathbb{R}^d$ with rank n, a parameter $s > 0$, a prime modulus $q \geq 10$, and a shift vector $\boldsymbol{c} \in \mathbb{R}^d$, let $\boldsymbol{x} \sim D_{\mathcal{L},s,\boldsymbol{c}}$ and $\boldsymbol{t} \sim \mathcal{L}^*/(q\mathcal{L}^*)$. Then, the distribution $(\boldsymbol{t}, \langle \boldsymbol{x}, \boldsymbol{t} \rangle \bmod q)$ is within statistical distance at most $2\varepsilon + 2^{3-k/2}$ of the uniform distribution over $\mathcal{L}^*/(q\mathcal{L}^*) \times \mathbb{Z}_q$, provided that*

$$s \geq 2q^{1/k} \cdot \eta_\varepsilon^{(k)}(\mathcal{L})$$

for $\varepsilon \in (0, 1/2)$.

Proof. Let $\mathcal{L}' \subseteq \mathcal{L}$ be a rank-k sublattice such that $s \geq 2q^{1/k}\eta_\varepsilon(\mathcal{L}')$, which is guaranteed by the fact that $s \geq 2q^{1/k}\eta_\varepsilon^{(k)}(\mathcal{L})$. Notice that by replacing $\mathcal{L}'$ with $\mathcal{L} \cap \operatorname{span}(\mathcal{L}')$, we may assume without loss of generality that $\mathcal{L}'$ is a primitive sublattice.

Next, notice that because $\mathcal{L}'$ is primitive, there exists some $\mathcal{L}''$ such that every vector in $\mathcal{L}$ can be written uniquely as a sum of a vector in $\mathcal{L}'$ and a vector in $\mathcal{L}''$. In particular, we can write $\boldsymbol{x} \sim D_{\mathcal{L},s,\boldsymbol{c}}$ as $\boldsymbol{x}' + \boldsymbol{x}''$. And, notice that conditioned on $\boldsymbol{x}''$, we have that $\boldsymbol{x}' \sim D_{\mathcal{L}',s,\boldsymbol{c}'}$, where $\boldsymbol{c}' := \boldsymbol{c} - \boldsymbol{x}''$. It therefore suffices to show that $(\boldsymbol{t}, \langle \boldsymbol{x}', \boldsymbol{t} \rangle \bmod q)$ is statistically close to uniform for $\boldsymbol{x}' \sim D_{\mathcal{L}',s,\boldsymbol{c}'}$ for any $\boldsymbol{c}' \in \mathbb{R}^d$. (In other words, we show that the result holds even after conditioning on $\boldsymbol{x}''$.) And, we can write $\boldsymbol{t} = \boldsymbol{t}' + \boldsymbol{t}''$ where $\boldsymbol{t}'' \in (\mathcal{L}')^\perp$ and $\boldsymbol{t}' \sim (\mathcal{L}')^*/(q(\mathcal{L}')^*)$. So, it suffices to show that with high probability over $\boldsymbol{t}' \sim (\mathcal{L}')^*/(q(\mathcal{L}')^*)$, $\langle \boldsymbol{x}', \boldsymbol{t} \rangle \bmod q$ is nearly uniformly random modulo q.

To that end, notice that by Corollary 2.5, the set

$$\mathcal{L}'_{\boldsymbol{t}'} := \{\boldsymbol{y} \in \mathcal{L}' \ : \ \langle \boldsymbol{t}', \boldsymbol{y} \rangle = 0 \bmod q\}$$

is a full-rank sublattice of $\mathcal{L}'$. Notice that $\boldsymbol{t}' \neq \boldsymbol{0}$ except with probability q^{-k}, and (because q is prime) this implies that for every i, the set $\{\boldsymbol{y} \in \mathcal{L}' \ : \ \langle \boldsymbol{t}', \boldsymbol{y} \rangle = i \bmod q\}$ is a distinct coset of $\mathcal{L}'_{\boldsymbol{t}'}$.[13] It therefore suffices to show that with high probability over $\boldsymbol{t}'$, the distribution of $\boldsymbol{x}' \sim D_{\mathcal{L}',s,\boldsymbol{c}'}$ is statistically close to uniform modulo $\mathcal{L}'_{\boldsymbol{t}'}$.

Indeed, Corollary 2.5 tells us that $\eta_{\varepsilon'}(\mathcal{L}'_{\boldsymbol{t}'}) \leq 2q^{1/k} \cdot \eta_\varepsilon(\mathcal{L}') \leq s$ where $\varepsilon' := \varepsilon + 2^{1-k/2}$ except with probability $2^{-k/2}$ over $\boldsymbol{t}'$. It then follows from Lemma 2.2 that $\boldsymbol{x}' \sim D_{\mathcal{L}',s,\boldsymbol{c}'}$ is within statistical distance $2\varepsilon' = 2\varepsilon + 2^{2-k/2}$ to uniform modulo $\mathcal{L}'_{\boldsymbol{t}'}$.

Combining all of this together, we achieve a final statistical distance bound of $2\varepsilon + 2^{2-k/2} + 2^{-k/2} + q^{-k} < 2\varepsilon + 2^{3-k/2}$, as needed. □

2.4 From lattices to quadratic forms

Instead of working with lattices $\mathcal{L}$ and their bases $\boldsymbol{B}$ directly, we will work with *quadratic forms* $\boldsymbol{Q}$. Formally, a quadratic form $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ is a positive-definite matrix—i.e., a symmetric matrix with all eigenvalues strictly positive. One can associate to every basis $\boldsymbol{B} \in \mathbb{R}^{d \times n}$ an associated quadratic form $\boldsymbol{Q} := \boldsymbol{B}^\top \boldsymbol{B} \in \mathcal{S}_n^{>0}(\mathbb{R})$ (known as the Gram matrix).

[13] Namely, if $\boldsymbol{y} \in \mathcal{L}'$ is such that $\langle \boldsymbol{t}', \boldsymbol{y} \rangle = 1 \bmod q$, then this coset is precisely $\mathcal{L}'_{\boldsymbol{t}'} + i\boldsymbol{y}$. Such a $\boldsymbol{y}$ exists because $\boldsymbol{t}' \neq \boldsymbol{0}$ and q is prime.

We can think of such a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$ as acting on the *coordinates* $\boldsymbol{z} \in \mathbb{Z}^n$ of a lattice vector $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{z} \in \mathcal{L}$. Indeed, notice that $\|\boldsymbol{y}\| = \|\boldsymbol{z}\|_{\boldsymbol{Q}}$, where we define

$$\|\boldsymbol{z}\|_{\boldsymbol{Q}} := (\boldsymbol{z}^\top \boldsymbol{Q} \boldsymbol{z})^{1/2} .$$

So, the quadratic form $\boldsymbol{Q}$ allows us to recover the norm $\|\boldsymbol{z}\|_{\boldsymbol{Q}}$ of a lattice vector $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{z}$ given its coordinates $\boldsymbol{z} \in \mathbb{Z}^n$ in some basis $\boldsymbol{B}$, without making direct use of the basis.

Indeed, one can think of the quadratic form $\boldsymbol{Q} := \boldsymbol{B}^\top \boldsymbol{B}$ as representing the basis $\boldsymbol{B}$ "modulo orthogonal transformations" (which are precisely the isometries of the Euclidean norm), since $\boldsymbol{B}_1^\top \boldsymbol{B}_1 = \boldsymbol{B}_2^\top \boldsymbol{B}_2$ if and only if there exists an orthogonal transformation $\boldsymbol{O} \in \mathsf{O}_d(\mathbb{R})$ such that $\boldsymbol{O}\boldsymbol{B}_1 = \boldsymbol{B}_2$. One can also recover a basis $\boldsymbol{B}$ from a quadratic form $\boldsymbol{Q}$, e.g., via the Cholesky decomposition—although, of course, one will only recover *some* basis $\boldsymbol{B}'$ such that $(\boldsymbol{B}')^\top \boldsymbol{B}' = \boldsymbol{Q}$, which might be an orthogonal transformation of the original basis $\boldsymbol{B}$ used to generate $\boldsymbol{Q}$.

One can then dispense with the lattice entirely and instead simply consider the quadratic form $\boldsymbol{Q}$ as representing a norm $\|\boldsymbol{z}\|_{\boldsymbol{Q}}$ on $\boldsymbol{z} \in \mathbb{Z}^n$. This is quite convenient computationally because it means that we only ever work with integer vectors $\boldsymbol{z} \in \mathbb{Z}^n$, rather than lattice vectors $\boldsymbol{y} \in \mathcal{L}$, which might have real coordinates. (At a more technical level, working with the quadratic form $\boldsymbol{Q}$ allows us to avoid working with orthogonal transformations $\boldsymbol{O}\mathcal{L}$ of lattices $\mathcal{L}$, which would otherwise be necessary and lead us into thorny issues about how to represent such orthogonal transformations efficiently. See [BGPS23, Section 1.3] for more discussion.)

For simplicity (and more-or-less without loss of generality), we restrict our attention to *integral* quadratic forms $\boldsymbol{Q} \in \mathbb{Z}^{n \times n}$.

We can translate some of the notions from Sections 2.1, 2.2 and 2.3 to this setting as follows.

For a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and $\boldsymbol{c} \in \mathbb{R}^n$, we write

$$\mathrm{dist}_{\boldsymbol{Q}}(\boldsymbol{c}, \mathbb{Z}^n) := \min_{\boldsymbol{z} \in \mathbb{Z}^n} \|\boldsymbol{c} - \boldsymbol{z}\|_{\boldsymbol{Q}} .$$

Notice that if $\boldsymbol{Q} = \boldsymbol{B}^\top \boldsymbol{B}$ for some basis $\boldsymbol{B}$ and $\mathcal{L} = \mathcal{L}(\boldsymbol{B})$, then

$$\mathrm{dist}_{\boldsymbol{Q}}(\boldsymbol{c}, \mathbb{Z}^n) = \mathrm{dist}(\boldsymbol{B}\boldsymbol{c}, \mathcal{L}) .$$

We will crucially rely on the simple fact that for any unimodular transformation $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$,

$$\mathrm{dist}_{\boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}}(\boldsymbol{c}, \mathbb{Z}^n) = \mathrm{dist}_{\boldsymbol{Q}}(\boldsymbol{U}\boldsymbol{c}, \mathbb{Z}^n) .$$

For a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, a parameter $s > 0$, and a vector $\boldsymbol{x} \in \mathbb{R}^n$, we define

$$\rho_{\boldsymbol{Q},s}(\boldsymbol{x}) := \exp(-\pi \|\boldsymbol{x}\|_{\boldsymbol{Q}}^2 / s^2) ,$$

and for $\boldsymbol{c} \in \mathbb{R}^n$, we write $D_{\boldsymbol{Q},s,\boldsymbol{c}}$ for the distribution over $\mathbb{Z}^n$ induced by this measure. In other words, for $\boldsymbol{z} \in \mathbb{Z}^n$,

$$\Pr_{\boldsymbol{X} \sim D_{\boldsymbol{Q},s,\boldsymbol{c}}}[\boldsymbol{X} = \boldsymbol{z}] := \frac{\rho_{\boldsymbol{Q},s}(\boldsymbol{z} - \boldsymbol{c})}{\rho_{\boldsymbol{Q},s}(\mathbb{Z}^n - \boldsymbol{c})} .$$

Notice that $D_{\boldsymbol{Q},s,\boldsymbol{c}}$ is closely related to $D_{\mathcal{L},s,\boldsymbol{t}}$ (which hopefully justifies the abuse of notation). In particular, if $\mathcal{L} = \mathcal{L}(\boldsymbol{B})$ for some basis $\boldsymbol{B}$ and $\boldsymbol{Q} = \boldsymbol{B}^\top \boldsymbol{B}$, then

$$\boldsymbol{B} D_{\boldsymbol{Q},s,\boldsymbol{c}} = D_{\mathcal{L},s,\boldsymbol{t}}$$

for $\boldsymbol{t} := \boldsymbol{B}\boldsymbol{c}$. In other words, $D_{\boldsymbol{Q},s,\boldsymbol{c}}$ is the distribution of the *coordinates* of a sample from $D_{\mathcal{L},s,\boldsymbol{t}}$.

Theorem 2.9 ([GPV08, BLP+13]). *There is an efficient algorithm that takes as input a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, parameter $s \geq \sqrt{\log(n) \cdot \|\boldsymbol{Q}\|}$, and shift vector $\boldsymbol{c} \in \mathbb{R}^n$ and outputs a sample from $D_{\boldsymbol{Q},s,\boldsymbol{c}}$.*[14]

We will also need the following rather technical corollary, which follows immediately from combining Theorem 2.9 with the celebrated LLL algorithm [LLL82].

Corollary 2.10. *There is an efficient algorithm that takes as input a quadratic form $\boldsymbol{P} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ where $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q}\boldsymbol{U}$ for some (unknown) unimodular transformation $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ and quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, parameter $s^2 \geq 2^n\|\boldsymbol{Q}\|$, and shift vector $\boldsymbol{c} \in \mathbb{R}^n$ and outputs a sample from $D_{\boldsymbol{P},s,\boldsymbol{c}}$.*

We can also define the smoothing parameter of a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ for $\varepsilon > 0$ as the unique parameter $\eta_\varepsilon(\boldsymbol{Q})$ such that $\rho_{\boldsymbol{Q}^{-1},1/\eta_\varepsilon(\boldsymbol{Q})}(\mathbb{Z}^n) = 1 + \varepsilon$. Equivalently, if $\boldsymbol{Q} = \boldsymbol{B}^\top\boldsymbol{B}$ and $\mathcal{L} = \mathcal{L}(\boldsymbol{B})$, then $\eta_\varepsilon(\boldsymbol{Q}) = \eta_\varepsilon(\mathcal{L})$.

We will need the basic fact that for any $\varepsilon := \varepsilon(n) \in (0, 1/2)$,

$$\eta_\varepsilon(\boldsymbol{I}_n) = \sqrt{\log(n/\varepsilon)/\pi} + o(1)\ . \tag{1}$$

We also generalize this as in Section 2.3 to

$$\eta_\varepsilon^{(k)}(\boldsymbol{Q}) = \min_{\boldsymbol{U}} \eta_\varepsilon(\boldsymbol{U}^\top\boldsymbol{Q}\boldsymbol{U})$$

where the minimum is over $\boldsymbol{U} \in \mathbb{Z}^{n\times k}$ that have full rank. As before, we have

$$\eta_\varepsilon^{(k)}(\boldsymbol{Q}) = \eta_\varepsilon^{(k)}(\mathcal{L})\ .$$

Below, we simply rewrite some of the results from the previous sections in terms of quadratic forms.[15]

Lemma 2.11 (Quadratic form version of Lemma 2.2). *For any quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, full-rank $\boldsymbol{D} \in \mathbb{Z}^{n\times n}$, $\varepsilon \in (0, 1/2)$, parameter $s \geq \eta_\varepsilon(\boldsymbol{D}^\top\boldsymbol{Q}\boldsymbol{D})$, and shift vector $\boldsymbol{t} \in \mathbb{R}^n$, the distribution $D_{\boldsymbol{Q},s,\boldsymbol{t}} \bmod \boldsymbol{D}\mathbb{Z}^n$ is within statistical distance 2ε of the uniform distribution over $\mathbb{Z}^n/(\boldsymbol{D}\mathbb{Z}^n)$.*

Lemma 2.12 (Quadratic form version of Lemma 2.3). *For any quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and $\varepsilon \in (0, 1/2)$, we have that*

$$\eta_\varepsilon(\boldsymbol{Q}) \leq \sqrt{\|\boldsymbol{Q}\| \cdot \log(n/\varepsilon)}.$$

Theorem 2.13 (Quadratic form version of Theorem 2.6). *For any quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, vector $\boldsymbol{c} \in \mathbb{R}^n$, and $s \geq \eta_\varepsilon(\boldsymbol{Q})$ for some $\varepsilon \in (0, 1/3)$,*

$$\Pr_{\boldsymbol{x}\sim D_{\boldsymbol{Q},s,\boldsymbol{c}}}[\|\boldsymbol{x} - \boldsymbol{c}\|_{\boldsymbol{Q}} \geq s\sqrt{n}] < 2^{-n}\ .$$

[14] In fact, it suffices to take $s \geq \sqrt{\log(2n+4) \cdot \|\widetilde{\boldsymbol{Q}}\|/\pi}$, where $\|\widetilde{\boldsymbol{Q}}\| \leq \|\boldsymbol{Q}\|$ is the maximal squared length of a Gram-Schmidt vector of a basis $\boldsymbol{B}$ such that $\boldsymbol{B}^\top\boldsymbol{B} = \boldsymbol{Q}$. But, for our purposes this is unnecessary.

[15] The authors recognize that it is rather unfortunate to write so many things twice. We did so because we found the proofs and intuition behind these results to be far simpler when they are presented as results about lattices, rather than forms. So, we presented and proved the results in terms of lattices above and here simply translate them into the form that we will need going forward.

Lemma 2.14 (Quadratic form version of Lemma 2.7)**.** *For any quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, parameter $s > 0$, vector $\boldsymbol{w} \in \mathbb{R}^n$, and $r > 0$,*

$$\Pr_{\boldsymbol{z} \sim D_{\boldsymbol{Q},s}}[\langle \boldsymbol{z}, \boldsymbol{w} \rangle \geq r] \leq \exp(-\pi r^2/(s^2 \|\boldsymbol{w}\|^2_{\boldsymbol{Q}^{-1}})).$$

Furthermore,

$$\Pr_{\boldsymbol{z} \sim D_{\boldsymbol{Q},s}}[|\langle \boldsymbol{z}, \boldsymbol{w} \rangle| \geq r] \leq 2\exp(-\pi r^2/(s^2 \|\boldsymbol{w}\|^2_{\boldsymbol{Q}^{-1}})).$$

Theorem 2.15 (Quadratic form version of Theorem 2.8)**.** *For any quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, integer $1 \leq k \leq n$, a prime modulus $q \geq 10$, parameter $s \geq 2q^{1/k} \cdot \eta_\varepsilon^{(k)}(\boldsymbol{Q})$ for some $\varepsilon \in (0, 1/2)$, and shift vector $\boldsymbol{c} \in \mathbb{R}^n$, let $\boldsymbol{z} \sim D_{\boldsymbol{Q},s,\boldsymbol{c}}$ and $\boldsymbol{t} \sim \mathbb{Z}_q^n$. Then, the distribution $(\boldsymbol{t}, \langle \boldsymbol{z}, \boldsymbol{t} \rangle \bmod q)$ is within statistical distance at most $2\varepsilon + 2^{3-k/2}$ of the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

For completeness, we also recall that for a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we have the following useful variant of the Cauchy-Schwarz inequality:

$$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \leq \|\boldsymbol{x}\|_{\boldsymbol{Q}} \|\boldsymbol{y}\|_{\boldsymbol{Q}^{-1}} \,. \tag{2}$$

(Note the inverse in the second subscript.) This is a special case of the Cauchy-Schwarz inequality for *dual* norms. One can also derive it directly from the classical Cauchy-Schwarz inequality by recalling that there exists a *symmetric* (in fact, positive definite) "matrix square root" $\boldsymbol{Q}^{1/2} \in \mathbb{R}^{n \times n}$ such that $(\boldsymbol{Q}^{1/2})^2 = \boldsymbol{Q}$. Then,

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{Q}^{1/2} \boldsymbol{Q}^{-1/2} \boldsymbol{y} = \langle \boldsymbol{Q}^{1/2} \boldsymbol{x}, \boldsymbol{Q}^{-1/2} \boldsymbol{y} \rangle$$

(where we have used the symmetry of $\boldsymbol{Q}^{1/2}$ in the second equality), and the Cauchy-Schwarz inequality implies that

$$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \leq \|\boldsymbol{Q}^{1/2} \boldsymbol{x}\| \|\boldsymbol{Q}^{-1/2} \boldsymbol{y}\| = \|\boldsymbol{x}\|_{\boldsymbol{Q}} \|\boldsymbol{y}\|_{\boldsymbol{Q}^{-1}} \,.$$

2.5 The Lattice Isomorphism Problem

Two lattices $\mathcal{L}_0, \mathcal{L}_1 \subset \mathbb{R}^n$ are *isomorphic* if there is an orthogonal transformation $O \in \mathsf{O}_n(\mathbb{R})$ such that $O\mathcal{L}_0 = \mathcal{L}_1$. Equivalently, if $\mathcal{L}_0 = \mathcal{L}(\boldsymbol{B}_0)$ and $\mathcal{L}_1 = \mathcal{L}(\boldsymbol{B}_1)$, then $\mathcal{L}_0$ is isomorphic to $\mathcal{L}_1$ if and only if there is an orthogonal transformation $\boldsymbol{O} \in \mathsf{O}_n(\mathbb{R})$ and a unimodular transformation $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ such that $\boldsymbol{O}\boldsymbol{B}_0\boldsymbol{U} = \boldsymbol{B}_1$.

We can remove the orthogonal transformation entirely by considering the quadratic forms $\boldsymbol{Q} := \boldsymbol{B}_0^\top \boldsymbol{B}_0$ and $\boldsymbol{P} := \boldsymbol{B}_1^\top \boldsymbol{B}_1$. Indeed, we see that $\mathcal{L}_0$ and $\mathcal{L}_1$ are isomorphic if and only if there exists a unimodular $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ such that $\boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U} = \boldsymbol{P}$.

It is easy to see that this is an equivalence relation, and we therefore say that two quadratic forms $\boldsymbol{Q}$ and $\boldsymbol{P}$ are *equivalent* and write $\boldsymbol{Q} \equiv \boldsymbol{P}$ if there exists $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ such that $\boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U} = \boldsymbol{P}$. And, we write

$$[\boldsymbol{Q}] := \{\boldsymbol{P} \;:\; \boldsymbol{P} \equiv \boldsymbol{Q}\}$$

for the equivalence class.

This leads naturally to the following computational problem.

Definition 2.16. *The (decision) Lattice Isomorphism Problem* LIP *is the computational problem in which the input is two quadratic forms $\boldsymbol{Q}, \boldsymbol{P} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and the goal is to determine whether $\boldsymbol{Q} \equiv \boldsymbol{P}$.*

However, we do not know how to build cryptography directly from the lattice isomorphism problem.

A recent line of work [DvW22, BGPS23, dCBZI+24, ARW24, BMM25, JWL+25a, LRvW25] has shown how to build cryptography from the following variants of this problem,[16] which one can think of as special cases of LIP in which $\boldsymbol{Q}$ is *fixed* to $\boldsymbol{Q}_0$ and we are promised that either $\boldsymbol{P} \equiv \boldsymbol{Q}_0$ or $\boldsymbol{P} \equiv \boldsymbol{Q}_1$. We present here both a worst-case version and an average-case version. (We will eventually observe that there is a worst-case to average-case reduction.)

Definition 2.17. *For $\boldsymbol{Q}_0 := \boldsymbol{Q}_0(n), \boldsymbol{Q}_1 := \boldsymbol{Q}_1(n) \in \mathcal{S}_n^{>0}(\mathbb{Z})$ such that $\boldsymbol{Q}_0 \not\equiv \boldsymbol{Q}_1$, the associated worst-case distinguishing Lattice Isomorphism Problem* $\mathrm{wcLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ *is the promise problem in which the input is $\boldsymbol{P} \in [\boldsymbol{Q}_0] \cup [\boldsymbol{Q}_1]$. The goal is to output 0 if $\boldsymbol{P} \equiv \boldsymbol{Q}_0$ and 1 if $\boldsymbol{P} \equiv \boldsymbol{Q}_1$.*

And, for any randomized algorithm $\mathcal{D}$ that takes as input $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and outputs $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$, the associated average-case distinguishing Lattice Isomorphism Problem $\mathcal{D}\text{-}\mathrm{acLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ *is the promise problem in which the input is $\boldsymbol{P} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ sampled as follows. Let $b \sim \{0,1\}$ be a uniformly random bit, and let $\boldsymbol{U} \leftarrow \mathcal{D}(\boldsymbol{Q}_b)$. Finally, let $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q}_b \boldsymbol{U}$. The goal is to output b.*

We say $\mathcal{D}\text{-}\mathrm{acLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ *is hard if no polynomial-time algorithm solves* $\mathcal{D}\text{-}\mathrm{acLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ *with probability non-negligibly larger than $1/2$.*

Intuitively, if $\mathcal{D}\text{-}\mathrm{acLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ is hard and we sample a secret key as $\boldsymbol{U} \leftarrow \mathcal{D}(\boldsymbol{Q}_0)$ and a public key as $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q}_0 \boldsymbol{U}$, then, from an adversary's perspective, $\boldsymbol{P}$ is indistinguishable from some $\boldsymbol{P}'$ with $\boldsymbol{P}' = (\boldsymbol{U}')^\top \boldsymbol{Q}_1 \boldsymbol{U}'$. So, if we can argue that the geometry of $\boldsymbol{Q}_1$ is such that our cryptographic scheme is necessarily secure when $\boldsymbol{P}$ is replaced by $\boldsymbol{P}'$ (e.g., perhaps encryptions of zero have nearly the same distribution as encryptions of one in this case), then we can argue that our scheme is secure under the assumption that $\mathcal{D}\text{-}\mathrm{acLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ is hard.

It remains to make a good choice of the sampling algorithm $\mathcal{D}$. This is (the quadratic form version of) the classical question of "how best to sample a bad basis for a lattice"—i.e., a basis that somehow does not reveal the geometry of the lattice. See, e.g., [BM21]. However, Haviv and Regev implicitly defined a distribution that is in a certain sense "as bad as possible" [HR14], and the two works [DvW22, BGPS23] then made this explicit. Roughly speaking, there is really a family of distributions with this property, which are parameterized by a Gaussian parameter $s \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}$ and an algorithm $\mathcal{A}$ that takes as input Gaussian samples $\boldsymbol{Z} \sim D_{\boldsymbol{Q},s}^m$ for $m \gg n$ and "reduces the degenerate form $\boldsymbol{Z}^\top \boldsymbol{Q} \boldsymbol{Z}$ into a non-degenerate form in a suitably nice way." E.g., $\mathcal{A}$ can be implemented using the LLL algorithm [LLL82] as in [BGPS23] or using the Hermite normal form as in [DvW22]. Either way, this results in a unimodular transformation $\boldsymbol{U}$ such that $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}$ has $\|\boldsymbol{P}\| \lesssim ns^2$ (with high probability). In particular, $\boldsymbol{P}$ will be such that one can sample efficiently from $D_{\boldsymbol{P},r}$ for $r \geq \sqrt{n \log n} s$ using Theorem 2.9.

And, one can show that as long as, say, $s \geq 2^{n/2} \max\{\sqrt{\|\boldsymbol{Q}_0\|}, \sqrt{\|\boldsymbol{Q}_1\|}\}$, this yields a distribution that is as hard as the worst-case distribution. (The proof observes that given *any* $\boldsymbol{Q}' \in [\boldsymbol{Q}_b]$, one can efficiently sample $\boldsymbol{P}'$ that is distributed identically to $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q}_b \boldsymbol{U}$ for $\boldsymbol{U} \sim \mathcal{D}_s(\boldsymbol{Q}_b)$.) We record all of this in the following theorem.

Theorem 2.18 ([HR14, DvW22, BGPS23])**.** *There is an efficient algorithm $\mathcal{D}_s$ that takes as input a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ with $\|\boldsymbol{Q}\| \leq s^2/\log n$ and outputs a unimodular matrix $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ with the following properties. Let $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}$ for $\boldsymbol{U} \leftarrow \mathcal{D}_s(\boldsymbol{Q})$.*

1. *$\|\boldsymbol{P}\| \leq \mathrm{poly}(n) s^2$ except with negligible probability.*

[16] See also [DPPvW23, BBD+25], which work with more structured variants of these problems.

2. *There is an efficient algorithm* $\mathcal{A}$ *that takes as input* $\boldsymbol{P}$ *and* $r \geq \sqrt{n \log n} \cdot s$ *and outputs a sample from* $D_{\boldsymbol{P},r}$.

3. *For any* $\boldsymbol{Q}_0, \boldsymbol{Q}_1 \in \mathcal{S}_n^{>0}(\mathbb{Z})$ *and any* $s \geq 2^{n/2} \max\{\sqrt{\|\boldsymbol{Q}_0\|}, \sqrt{\|\boldsymbol{Q}_1\|}\}$*, there is an efficient reduction from* $\text{wcLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ *to* $\mathcal{D}_s\text{-acLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$*. So, with these parameters,* $\text{wcLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ *is hard if and only if* $\mathcal{D}_s\text{-acLIP}^{\boldsymbol{Q}_0,\boldsymbol{Q}_1}$ *is hard.*

2.6 Identity-based encryption

An *identity-based encryption* scheme is a four-tuple of polynomial-time (randomized) algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with the following behavior.

- The *setup algorithm* Setup takes as input the security parameter 1^n and outputs a *master secret key* msk and a *master public key* mpk.
- The *key-generation algorithm* KeyGen takes as input the master secret key msk and an *identity* $\mathsf{id} \in \{0,1\}^*$ and outputs a secret key $\mathsf{sk}_{\mathsf{id}}$.
- The *encryption algorithm* Enc takes as input the master public key mpk, an identity $\mathsf{id} \in \{0,1\}^*$, and a *plaintext* bit $\mu \in \{0,1\}$ and outputs a *ciphertext* c.
- The *decryption algorithm* Dec takes as input a secret key $\mathsf{sk}_{\mathsf{id}}$ and a ciphertext c and outputs a plaintext bit $\mu \in \{0,1\}$.

Such a scheme is *correct* if for any security parameter n, identity $\mathsf{id} \in \{0,1\}^*$, and bit $\mu \in \{0,1\}$, we have

$$\Pr_{\substack{(\mathsf{msk},\mathsf{mpk}) \leftarrow \mathsf{Setup}(1^n) \\ \mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk},\mathsf{id})}} [\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mu)) \neq \mu] < \text{negl}(n) \ .$$

In words, if one encrypts μ using the master public key mpk and the identity id and if $\mathsf{sk}_{\mathsf{id}}$ was generated by $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$, then decrypting the resulting ciphertext using $\mathsf{sk}_{\mathsf{id}}$ will yield μ with high probability. One can think of this as "a public-key encryption scheme where your public key can be any string $\mathsf{id} \in \{0,1\}^*$" (e.g., your name or your e-mail address).

Definition 2.19. Full security *of an identity-based encryption scheme* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is defined in terms of the following game between an adversary and a challenger. At the start of the game, the adversary and the challenger take as input the security parameter in unary* 1^n*. They then interact as follows.*

1. *The challenger computes* $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^n)$ *and sends* mpk *to the adversary.*
2. *The adversary makes as many adaptive* identity queries $\mathsf{id}_1, \ldots, \mathsf{id}_{\ell_1}$ *as she likes. In other words, for* $i = 1, \ldots, \ell_1$ *with* $\ell_1 \leq \text{poly}(n)$ *chosen by the adversary, the adversary sends* $\mathsf{id}_i \in \{0,1\}^*$ *to the challenger, and the challenger responds with* $\mathsf{sk}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}_i)$.
3. *The adversary sends the* challenge identity $\mathsf{id}^* \in \{0,1\}^* \setminus \{\mathsf{id}_1, \ldots, \mathsf{id}_{\ell_1}\}$ *to the challenger.*
4. *The challenger samples* $\mu^* \sim \{0,1\}$ *and sends* $c^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*, \mu^*)$ *to the adversary.*
5. *The adversary makes as many additional adaptive identity queries* $\mathsf{id}_{\ell_1+1}, \ldots, \mathsf{id}_{\ell_1+\ell_2}$ *as she likes with the restriction that* $\mathsf{id}_i \neq \mathsf{id}^*$.

6. The adversary sends $\mu \in \{0,1\}$ to the challenger and wins if $\mu = \mu^*$.

An identity-based encryption scheme is then fully secure *if for any probabilistic* poly(n)*-time adversary there exists a negligible function $\varepsilon(n)$ such that the probability that the adversary wins the above game is bounded by $1/2 + \varepsilon(n)$.*

2.7 Leveled fully homomorphic encryption

For an integer $L = L(n) \geq 1$, a *level L fully homomorphic encryption scheme* is a four-tuple of polynomial-time (randomized) algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ with the following behavior.

- The *key-generation algorithm* Gen takes as input the security parameter 1^n and outputs a *secret key* sk and a *public key* pk.
- The *encryption algorithm* Enc takes as input the public key pk and a *plaintext* bit $\mu \in \{0,1\}$ and outputs a *ciphertext* c.
- The *decryption algorithm* Dec takes as input the secret key sk and a ciphertext c and outputs a plaintext bit $\mu \in \{0,1\}$.
- The *homomorphic evaluation algorithm* Eval takes as input the public key pk, a circuit $f : \{0,1\}^\ell \to \{0,1\}$ consisting of $\oplus$ and $\times$ gates, and a list $c_1, \ldots, c_\ell$ of ciphertexts, and outputs a ciphertext c^*.

We sometimes abuse notation and write $C \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$ for $M = (\mu_1, \ldots, \mu_\ell) \in \{0,1\}^\ell$ to mean that $C = (c_1, \ldots, c_\ell)$ where $c_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \mu_i)$.

Such a scheme is *correct* if for any positive integer n, any plaintext bits $M \in \{0,1\}^\ell$, and any $(\oplus, \times)$-circuit $f : \{0,1\}^\ell \to \{0,1\}$ with depth at most $L(n)$, we have

$$\Pr_{(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}(1^n)} [C \leftarrow \mathsf{Enc}(\mathsf{pk}, M),\ \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, C)) \neq f(\mu_1, \ldots, \mu_\ell)] < \mathrm{negl}(n) \ .$$

In other words, if one encrypts $\mu_1, \ldots, \mu_\ell$ using the public key, homomorphically evaluates the circuit f on the resulting ciphertexts, and then runs the decryption algorithm on the result using the secret key sk, the result decrypts to $f(\mu_1, \ldots, \mu_\ell)$ except with negligible probability.

Such a scheme is *compact* if the bit length of the output of $\mathsf{Eval}(\mathsf{pk}, f, C)$ is bounded by some polynomial in the bit length of pk (independent of f and C).[17]

Such a scheme is *secure* if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is semantically secure as a public-key encryption scheme. That is, it is secure if for any probabilistic polynomial-time adversary $\mathcal{A}$,

$$\Pr_{(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}(1^n), \mu \sim \{0,1\}} [\mathcal{A}(1^n, \mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mu)) = \mu] < \frac{1}{2} + \mathrm{negl}(n) \ .$$

[17] Intuitively, we insist on compact schemes because we want "the evaluation algorithm to do most of the work to evaluate the circuit f." If we did not insist on this, then the evaluation algorithm could simply, e.g., include the full list of ciphertexts C and the entire circuit f as part of c^*, and the decryption algorithm could compute $f(\mu_1, \ldots, \mu_\ell)$ directly from C and f.

3 A public-key encryption scheme that we will use to build IBE

We first describe a public-key encryption scheme that we will use to build IBE. Our scheme has as parameters a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, a parameter $s := s(n) \geq \sqrt{\log n \|\boldsymbol{Q}\|}$ (which controls the size of the public quadratic form $\boldsymbol{P}$), a noise parameter $r := r(n) \geq \sqrt{n \log n} \cdot s/q$, a prime discretization parameter $q := q(n) \geq 10$, and a distance bound $\rho := \rho(n) > 0$.

We require that $\boldsymbol{Q}$ is such that there is some efficient algorithm that outputs $\boldsymbol{t}' \sim \mathbb{T}_q^n$ together with $\boldsymbol{y}' \in \mathbb{Z}^n$ such that $\|\boldsymbol{y}' - \boldsymbol{t}'\|_{\boldsymbol{Q}^{-1}} \leq \rho$.[18] We write $h := (q+1)/(2q) \in \mathbb{T}_q$.

- $\mathsf{Gen}(1^n)$: Sample $\boldsymbol{U} \leftarrow \mathcal{D}_s(\boldsymbol{Q}) \in \mathrm{GL}_n(\mathbb{Z})$, and compute $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}$. Sample $(\boldsymbol{t}', \boldsymbol{y}') \in \mathbb{T}_q^n \times \mathbb{Z}^n$ such that $\|\boldsymbol{t}' - \boldsymbol{y}'\|_{\boldsymbol{Q}^{-1}} \leq \rho$ as above and set $\boldsymbol{t} := \boldsymbol{U}^\top \boldsymbol{t}' \bmod 1$ and $\boldsymbol{y} := \boldsymbol{U}^\top \boldsymbol{y}' + \boldsymbol{t} - \boldsymbol{U}^\top \boldsymbol{t}' \in \mathbb{Z}^n$ (so that $\|\boldsymbol{t} - \boldsymbol{y}\|_{\boldsymbol{P}^{-1}} = \|\boldsymbol{t}' - \boldsymbol{y}'\|_{\boldsymbol{Q}^{-1}} \leq \rho$). Output $\mathsf{sk} := \boldsymbol{y}$ and $\mathsf{pk} := (\boldsymbol{P}, \boldsymbol{t})$.
- $\mathsf{Enc}(\mathsf{pk}, \mu \in \{0,1\})$: Sample $\boldsymbol{e} \sim D_{\boldsymbol{P}, rq}/q \in \mathbb{Z}^n/q$ using algorithm $\mathcal{A}$ in Theorem 2.18. Compute $\boldsymbol{c} := \boldsymbol{e} \bmod 1 \in \mathbb{T}_q^n$ and $d := \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle + \mu h \bmod 1 \in \mathbb{T}_q$. Output $(\boldsymbol{c}, d)$.
- $\mathsf{Dec}(\mathsf{sk}, (\boldsymbol{c}, d))$: Compute $d + \langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle \bmod 1$. Output 1 if the result is in the interval $[1/4, 3/4]$, and 0 otherwise.

This scheme is effectively a lattice-isomorphism variant of the dual-Regev encryption scheme introduced in [GPV08]. It is also quite similar to the simplest encryption scheme presented in [BMM25], and to the key-encapsulation mechanism proposed in [DvW22]. The main difference from [DvW22] is that we avoid using a generic extractor like in [DvW22]. Instead, we show directly (using Theorem 2.15) that $\langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle \bmod 1$ is pseudorandom, even conditioned on $\boldsymbol{t} \neq \boldsymbol{0}$ and $\boldsymbol{c}$. ([BMM25] also use a linear extractor, but they must switch moduli to make their proof work.)

To understand roughly why this scheme is correct, notice that because $\boldsymbol{e} - \boldsymbol{c} \in \mathbb{Z}^n$ and $\boldsymbol{y} \in \mathbb{Z}^n$, we have that

$$d = \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle + \mu h = \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle + \mu h \bmod 1 \,.$$

Therefore,

$$d + \langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle = \mu h + \langle \boldsymbol{e}, \boldsymbol{t} - \boldsymbol{y} \rangle \bmod 1 \,.$$

Since $\|\boldsymbol{t} - \boldsymbol{y}\|_{\boldsymbol{P}^{-1}}$ is small and $\|\boldsymbol{e}\|_{\boldsymbol{P}}$ is small, we expect $\langle \boldsymbol{e}, \boldsymbol{t} - \boldsymbol{y} \rangle$ to be small (this holds by the Cauchy-Schwarz inequality in Equation (2), although we use a more sophisticated argument below), and therefore that $d + \langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle \approx \mu h \approx \mu/2 \bmod 1$. We prove this formally in Theorem 3.1.

On the other hand, imagine that $\boldsymbol{P}$ is indistinguishable from some $\boldsymbol{P}'$ that has the following property. If we were to sample $\boldsymbol{e} \sim D_{\boldsymbol{P}', rq}/q$ (rather than from $D_{\boldsymbol{P}, rq}/q$), then even conditioned on $\boldsymbol{c} = \boldsymbol{e} \bmod 1$, the vector $\boldsymbol{e} - \boldsymbol{c}$ still has high entropy. Then, one might roughly hope that the ciphertext $(\boldsymbol{c}, \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle + \mu h \bmod 1)$ information-theoretically hides the plaintext μ. This is effectively how we will prove security in Theorem 3.2, using the machinery from Section 2.3.

[18] In Section 4, the algorithm effectively directly samples $\boldsymbol{t}' \sim \mathbb{T}_q^n$ uniformly at random and then it samples $\boldsymbol{y}' \sim D_{\boldsymbol{Q}^{-1}, \rho/\sqrt{n}, \boldsymbol{t}'}$. But, in many similar schemes, a short $\boldsymbol{x}' \in \mathbb{Z}^n/q$ is sampled first and then $\boldsymbol{t}'$ is simply set to $\boldsymbol{x}' \bmod 1$ and $\boldsymbol{y}' = \boldsymbol{t}' - \boldsymbol{x}'$. We leave this unspecified here, since in this section, we are agnostic to how $\boldsymbol{y}'$ and $\boldsymbol{t}'$ are generated. (Of course, ρ must be at least the covering radius in order for this to be plausibly possible.) Indeed, security of our scheme in Section 4 will crucially rely on the fact that there is this alternative way to sample $\boldsymbol{t}'$ and $\boldsymbol{y}'$, effectively sampling $\boldsymbol{t}' - \boldsymbol{y}'$ first and then setting $\boldsymbol{t}' := \boldsymbol{t}' - \boldsymbol{y}' \bmod 1$. (A similar dynamic occurs in [GPV08].) Of course, we could simply ask that the distribution of $\boldsymbol{t}'$ is statistically close to uniform, but in our applications we will have true uniformity.

3.1 Correctness

Theorem 3.1. *Let $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, $r := r(n) \geq \sqrt{n \log n} \cdot s/q$ and $\rho := \rho(n) > 0$ be as in the encryption scheme above. Then, for any $\mu \in \{0,1\}$,*

$$\Pr_{\substack{(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}(1^n) \\ (\boldsymbol{c},d) \leftarrow \mathsf{Enc}(\mathsf{pk},\mu)}} [\mathsf{Dec}(\mathsf{sk},(\boldsymbol{c},d)) \neq \mu] \leq 2\exp(-\pi/(25r^2\rho^2)) \ .$$

In particular, the encryption scheme is correct (i.e., has negligible chance of decryption failure) if $r\rho < o(1/\sqrt{\log n})$.

Proof. We have

$$\begin{aligned} d + \langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle &= \mu h + \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle + \langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle \\ &= \mu h + \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle + \langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle \\ &= \mu h + \langle \boldsymbol{e}, \boldsymbol{t} - \boldsymbol{y} \rangle \bmod 1 \ , \end{aligned}$$

where the second equality uses the fact that $\langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{y} \rangle = 0 \bmod 1$, as we discussed above.

Recall that we have $\|\boldsymbol{y}' - \boldsymbol{t}'\|_{\boldsymbol{Q}^{-1}} \leq \rho$ by assumption, which implies that $\|\boldsymbol{y} - \boldsymbol{t}\|_{\boldsymbol{P}^{-1}} \leq \rho$. Since $\boldsymbol{e} \sim D_{\boldsymbol{P},rq}/q$, it follows immediately from Lemma 2.14 that $|\langle \boldsymbol{e}, \boldsymbol{t} - \boldsymbol{y} \rangle| < 1/5$ except with probability at most $2\exp(-\pi/(25r^2\rho^2))$, and the result follows. □

3.2 Security

We now show the semantic security of this scheme. In fact, we show the slightly stronger fact that the ciphertexts are pseudorandom. We note that a very similar argument appeared in [DvW22]; the main difference is that we use Theorem 2.15 in order to avoid using an extractor like in [DvW22].

Theorem 3.2. *Suppose that there exists some $\boldsymbol{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$ such that for the parameters $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, $s := s(n) \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}$, $r := r(n) \geq \sqrt{n \log n} \cdot s/q$ and $q := q(n) \geq 10$ in the encryption scheme above, $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard and*

$$r \geq 2q^{1/k} \cdot \eta_\varepsilon^{(k)}(\boldsymbol{Q}')$$

for some $\varepsilon < \mathrm{negl}(n)$ and $k > \omega(\log(n))$. Then the joint distribution $(\boldsymbol{t}, \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle \bmod 1)$ is computationally indistinguishable from $(\boldsymbol{t}, u) \sim \mathbb{T}_q^n \times \mathbb{T}_q$, even conditioned on $\boldsymbol{c}$. In particular, the encryption scheme above is semantically secure.

Proof. Consider $\mathbf{Game}_1$ and $\mathbf{Game}_2$, where $\mathbf{Game}_1$ is the security game with respect to the original encryption scheme, and $\mathbf{Game}_2$ is almost identical but with public key drawn with respect to $\boldsymbol{Q}'$. By assumption, the two games are computationally indistinguishable. Now we argue that any adversary wins $\mathbf{Game}_2$ with negligible advantage.

Because $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ has $\det(\boldsymbol{U}) = \pm 1$ it is invertible mod q, and therefore the map $\boldsymbol{s} \mapsto \boldsymbol{U}^\top \boldsymbol{s} \bmod 1$ is a bijection on $\mathbb{T}_q^n$. Because we assume that $\boldsymbol{t}' \sim \mathbb{T}_q^n$ is uniformly random and $\boldsymbol{t} := \boldsymbol{U}^\top \boldsymbol{t}' \bmod 1$, we therefore have that $\boldsymbol{t}$ is also uniformly random in $\mathbb{T}_q^n$. Furthermore, conditioned on a fixed vector $\boldsymbol{c}$, the distribution of $\boldsymbol{e} - \boldsymbol{c}$ is precisely $D_{\boldsymbol{P}',r,-\boldsymbol{c}}$, where $\boldsymbol{P}' := \boldsymbol{U}^\top \boldsymbol{Q}' \boldsymbol{U}$. Using the fact that $\eta_\varepsilon^{(k)}(\boldsymbol{P}') = \eta_\varepsilon^{(k)}(\boldsymbol{Q}')$ and the assumption that $r \geq 2q^{1/k} \cdot \eta_\varepsilon^{(k)}(\boldsymbol{Q}')$, we therefore have that $(\boldsymbol{t}, \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle \bmod 1)$ is within statistical distance $2\varepsilon + 2^{3-k/2} < \mathrm{negl}(n)$ of uniformly random $(\boldsymbol{t}, u) \sim \mathbb{T}_q^n \times \mathbb{T}_q$ by Theorem 2.15 (applied with values scaled by a factor of q). □

4 Identity-based encryption

The scheme has the same parameters as the simpler scheme from Section 3. In particular, let $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ be a quadratic form, $s := s(n) \geq \sqrt{\|\boldsymbol{Q}\| \log n}$ a parameter that controls the size of $\boldsymbol{P}$, $r := r(n) \geq s\sqrt{n \log n}/q$ a noise parameter, $q := q(n) \geq 10$ a prime discretization parameter, and $\rho := \rho(n) > 0$ a distance bound.

In addition, we now also require that $\rho > \omega(\sqrt{\|\boldsymbol{Q}^{-1}\| \cdot n \log n})$. Furthermore, let $\mathcal{H} : \{0,1\}^* \to \mathbb{T}_q^n$ be a hash function modeled as a random oracle. We again write $h := (q+1)/(2q) \in \mathbb{T}_q$.

- $\mathsf{Setup}(1^n)$: Sample $\boldsymbol{U} \leftarrow \mathcal{D}_s(\boldsymbol{Q}) \in \mathrm{GL}_n(\mathbb{Z})$, and compute $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}$. Output $(\mathsf{mpk}, \mathsf{msk}) = (\boldsymbol{P}, \boldsymbol{U})$.
- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$: Compute $\boldsymbol{t} := \mathcal{H}(\mathsf{id}) \in \mathbb{T}_q^n$. Sample $\boldsymbol{y}' \sim D_{\boldsymbol{Q}^{-1}, \rho/\sqrt{n}, \boldsymbol{U}^{-\top}\boldsymbol{t}}$ (using the algorithm from Theorem 2.9). Compute $\boldsymbol{y} := \boldsymbol{U}^\top \boldsymbol{y}'$, and output $\mathsf{sk}_{\mathsf{id}} := \boldsymbol{y} \in \mathbb{Z}^n$.
- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mu \in \{0,1\})$: Compute $\boldsymbol{t} = \mathcal{H}(\mathsf{id})$. Sample $\boldsymbol{e} \sim D_{\boldsymbol{P}, rq}/q$ (using algorithm $\mathcal{A}$ from Theorem 2.18). Compute $\boldsymbol{c} := \boldsymbol{e} \bmod 1 \in \mathbb{T}_q^n$ and $d := \langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle + \mu h \bmod 1 \in \mathbb{T}_q$. Output $(\boldsymbol{c}, d)$.
- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, (\boldsymbol{c}, d))$: Compute $d + \langle \boldsymbol{c}, \boldsymbol{t} - \boldsymbol{y} \rangle \bmod 1$. Output 1 if the result is in $[1/4, 3/4]$, or 0 otherwise.

Notice that our new requirement that $\rho > \omega(\sqrt{\|\boldsymbol{Q}^{-1}\| n \log n})$ guarantees that we can run the algorithm in Theorem 2.9 to sample $\boldsymbol{y}' \sim D_{\boldsymbol{Q}^{-1}, \rho/\sqrt{n}, \boldsymbol{U}^{-\top}\boldsymbol{t}}$. (For that, we would "only" need $\rho \geq \sqrt{\|\boldsymbol{Q}^{-1}\| n \log n}$. But, later for security we will need the slightly stronger inequality $\rho > \omega(\sqrt{\|\boldsymbol{Q}^{-1}\| n \log n})$.)

More interestingly, notice that this scheme is closely related to the simpler public-key encryption scheme from Section 3. Indeed, one more-or-less recovers the scheme from Section 3 if one simply ignores the identity id and instead replaces $\boldsymbol{t} := \mathcal{H}(\mathsf{id})$ with a uniformly random element $\boldsymbol{t} \sim \mathbb{T}_q^n$ that is included with the public key. In particular, (by Theorem 2.13) we have that with high probability $\|\boldsymbol{y}' - \boldsymbol{t}'\|_{\boldsymbol{Q}^{-1}} < \rho$ where letting $\boldsymbol{t}' := \boldsymbol{U}^{-\top}\boldsymbol{t}$, so with high probability $\boldsymbol{y}$ and $(\boldsymbol{P}, \boldsymbol{t})$ satisfy the same relationship as the secret key and public key in Section 3. This is quite similar to the relationship between the GPV IBE scheme and the dual-Regev encryption scheme in [GPV08].

4.1 Correctness

Theorem 4.1. *Let $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, $r := r(n) \geq \sqrt{n \log n} \cdot s/q$ and $\rho := \rho(n) > \omega(\sqrt{\|\boldsymbol{Q}^{-1}\| n \log n})$ be as in the encryption scheme above. Then, for any $\mu \in \{0,1\}$,*

$$\Pr_{\substack{(\mathsf{msk},\mathsf{mpk}) \leftarrow \mathsf{Setup}(1^n) \\ \mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk},\mathsf{id})}} [\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mu)) \neq \mu] \leq 2\exp(-\pi/(25r^2\rho^2)) + 2^{-n}.$$

In particular, the encryption scheme is correct, provided that $r\rho < o(1/\sqrt{\log n})$.

Proof. Let $\boldsymbol{t}' := \boldsymbol{U}^{-\top}\boldsymbol{t}$ and notice that since $\boldsymbol{t}$ is a uniformly random element of $\mathbb{T}_q^n$, $\boldsymbol{t}' \bmod 1$ is also a uniformly random element of $\mathbb{T}_q^n$ (since $\det(\boldsymbol{U}) = \pm 1$ and therefore $\boldsymbol{U}$ is invertible modulo q). By Theorem 2.13, we have that $\|\boldsymbol{y}' - \boldsymbol{t}'\|_{\boldsymbol{Q}^{-1}} < \rho$ where $\boldsymbol{t}' := \boldsymbol{U}^{-\top}\boldsymbol{t}$ except with probability at most 2^{-n}. Conditioned on this event, we see that we are in the setting of Theorem 3.1, which tells us that the failure probability is at most $2\exp(-\pi/(25r^2\rho^2))$. The result then follows from the union bound. □

4.2 Full security

We next show full security of our IBE scheme. (Recall the definition of full security from Section 2.6.)

Theorem 4.2. *Suppose that there exists some $\boldsymbol{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$ such that for the parameters $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, $s := s(n) \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}$, $r := r(n) \geq \sqrt{n \log n} \cdot s/q$, prime $q := q(n) \geq 2^{n/2}$, and $\rho := \rho(n) > \omega(\sqrt{\|\boldsymbol{Q}^{-1}\| n \log n})$ in the encryption scheme above, $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard and*

$$r \geq 2q^{1/k} \cdot \eta_\varepsilon^{(k)}(\boldsymbol{Q}')$$

for some $\varepsilon < \mathrm{negl}(n)$ and $k > \omega(\log(n))$. Then the IBE scheme is fully secure.

Proof. We define a sequence of games, called $\mathbf{Game}_1, \mathbf{Game}_2$, and $\mathbf{Game}_3$. $\mathbf{Game}_1$ is the original full-security game against our IBE scheme (as in Definition 2.19). In particular, in $\mathbf{Game}_1$,

- $\boldsymbol{t}_i = \mathcal{H}(\mathsf{id}_i)$ is obtained from a random oracle $\mathcal{H}$ and is therefore uniformly random in $\mathbb{T}_q^n$; and
- $\boldsymbol{y}_i$ is obtained by sampling $\boldsymbol{y}_i' \leftarrow D_{\boldsymbol{Q}^{-1},\rho/\sqrt{n},U^{-\top}\boldsymbol{t}_i}$ and computing $\boldsymbol{y}_i = \boldsymbol{U}^\top \boldsymbol{y}_i'$.

In $\mathbf{Game}_2$, the adversary is required to tell the random oracle $\mathcal{H}$ whether its query id is equal to id^*. Formally, this can be done by replacing the random oracle $\mathcal{H}$ with a new random oracle $\mathcal{O}$ that takes as input $(b, \mathsf{id}) \in \{0,1\} \times \{0,1\}^*$, where we think of the bit b as indicating whether id is the unique identity that the adversary chooses as id^*. Specifically, the adversary loses this new game if either (1) it sends two queries of the form $(1, \mathsf{id}), (1, \mathsf{id}')$ where $\mathsf{id} \neq \mathsf{id}'$ (since we cannot have that $\mathsf{id} = \mathsf{id}^*$ and $\mathsf{id}' = \mathsf{id}^*$); or (2) it sends id^* to the challenger after having made the query $(0, \mathsf{id}^*)$. (As this is syntactically cumbersome, we do not write this formalism explicitly below.)

Claim 4.3. *For any polynomial-time adversary $\mathcal{A}$ in $\mathbf{Game}_1$ making at most Q queries, including queries to the random oracle $\mathcal{O}$ and identity queries (recall Definition 2.19, Item 2) with advantage $\delta = \delta(n) > 0$, there is a polynomial-time adversary $\mathcal{B}$ with advantage δ/Q in $\mathbf{Game}_2$.*

Proof. First, replace $\mathcal{A}$ by $\mathcal{A}'$ that makes no repeated oracle queries/identity queries and that precedes each query to the challenger (including id^*) by a random oracle query on the same id (if it has not already made this query). Notice that the number of random oracle queries made by $\mathcal{A}'$ is at most Q. Then, we construct an adversary $\mathcal{B}$ that first samples $i^* \sim [Q]$ corresponding to a guess of the index of the oracle query of id^*. $\mathcal{B}$ then simulates $\mathcal{A}'$. When $\mathcal{A}'$ makes the i^*th query, $\mathcal{B}$ declares the corresponding id to be id^* (i.e., sets $b = 1$ in that query). If $\mathcal{A}'$ chooses a different id to be id^*, $\mathcal{B}$ simply outputs a uniformly random bit (after sending any unused identity to the challenger as its challenge identity $(\mathsf{id}^*)'$). Otherwise, $\mathcal{B}$ outputs the bit that $\mathcal{A}'$ outputs. The probability of $\mathcal{B}$ winning is

$$\begin{aligned}
\Pr[\mathcal{B} \text{ wins } \mathbf{Game}_2] &= \frac{1}{2} \cdot \Pr[\mathcal{B} \text{ incorrectly guesses } i^*] + \Pr[\mathcal{B} \text{ correctly guesses } i^*] \cdot \Pr[\mathcal{A}' \text{ wins } \mathbf{Game}_1] \\
&= \frac{1}{2} \cdot \left(1 - \frac{1}{Q}\right) + \frac{1}{Q} \cdot \left(\frac{1}{2} + \delta\right) \\
&= \frac{1}{2} + \frac{\delta}{Q},
\end{aligned}$$

as needed, where we have used the fact that the random guess of $\mathcal{B}$ is independent of whether $\mathcal{A}'$ wins $\mathbf{Game}_1$. □

We define $\mathbf{Game}_3$ to be the same as $\mathbf{Game}_2$, except that for every random oracle query or identity query, if $\mathsf{id} \neq \mathsf{id}^*$, $\boldsymbol{t}_i$ and $\boldsymbol{y}_i$ are sampled as follows

- Sample $\boldsymbol{v}_i \sim D_{\boldsymbol{P}^{-1},\rho q/\sqrt{n}}/q$ using the algorithm from Corollary 2.10. (This works because $\rho q/\sqrt{n} \geq 2^{n/2}\sqrt{\|\boldsymbol{Q}^{-1}\|}$.)
- Compute $\boldsymbol{t}_i := \boldsymbol{v}_i \bmod 1 \in \mathbb{T}_q^n$, and $\boldsymbol{y}_i := \boldsymbol{t}_i - \boldsymbol{v}_i \in \mathbb{Z}^n$.

(If the adversary later makes a challenge query with the same id, the challenger responds with $\boldsymbol{y}_i$.) Notice that $\mathbf{Game}_3$ is well-defined because of the fact that the adversary in $\mathbf{Game}_2$ (and also $\mathbf{Game}_3$) is required to tell the challenger whether $\mathsf{id} = \mathsf{id}^*$.

Claim 4.4. *Any adversary $\mathcal{A}$ making at most Q queries with advantage δ in $\mathbf{Game}_2$ has advantage at least $\delta - 2nQ\exp(-\rho^2/(n\|\boldsymbol{Q}^{-1}\|)) > \delta - Q\,\mathrm{negl}(n)$ in $\mathbf{Game}_3$.*

Proof. By Lemma 2.12 we have that $\rho/\sqrt{n} \geq \eta_{\varepsilon'}(\boldsymbol{Q}^{-1}) = \eta_{\varepsilon'}(\boldsymbol{P}^{-1})$, where

$$\varepsilon' := n \cdot \exp(-\rho^2/(n\|\boldsymbol{Q}^{-1}\|)) < \mathrm{negl}(n)\ ,$$

since $\rho/\sqrt{n} \geq \omega(\sqrt{\|\boldsymbol{Q}^{-1}\|\log n})$. So, by Lemma 2.11, $\boldsymbol{t}_i = \boldsymbol{v}_i \bmod 1$ is within statistical distance $2\varepsilon'$ of a uniformly random vector $\boldsymbol{t}_i \sim \mathbb{T}_q^n$, which is the case in $\mathbf{Game}_2$ as we used a random oracle. And, conditioned on $\boldsymbol{t}_i$, the element $\boldsymbol{y}_i$ in both games has distribution exactly $D_{\boldsymbol{P}^{-1},\rho/\sqrt{n},\boldsymbol{t}_i}$. Thus the keys sampled in $\mathbf{Game}_3$ are statistically indistinguishable from the ones in $\mathbf{Game}_2$, and the result follows by taking a union bound over Q queries. □

Claim 4.5. *If there exists a polynomial-time adversary $\mathcal{A}$ that breaks $\mathbf{Game}_3$ with advantage δ, then there exists a polynomial-time adversary $\mathcal{B}$ that can break the encryption scheme in Section 3 with the same advantage δ.*

Proof. The adversary $\mathcal{B}$, which breaks the encryption scheme in Section 3, works as follows.

- $\mathcal{B}$ receives public key $\mathsf{pk} := (\boldsymbol{P}, \boldsymbol{t})$ and ciphertext $(\boldsymbol{c}, d) \leftarrow \mathsf{Enc}(\mathsf{pk}, \mu \in \{0,1\})$.
- $\mathcal{B}$ calls the adversary $\mathcal{A}$ with $\mathsf{mpk} := \boldsymbol{P}$. Whenever $\mathcal{A}$ requests a random oracle query for $\mathsf{id} \neq \mathsf{id}^*$ or an identity query, $\mathcal{B}$ forges the random oracle response $\boldsymbol{t}_i$ and the secret key $\boldsymbol{y}_i$ by the process described above. When $\mathcal{A}$ requests the random oracle query for id^*, $\mathcal{B}$ sends $\boldsymbol{t}$ (from its public key) to $\mathcal{A}$.
- When $\mathcal{A}$ sends id^* to the challenger, $\mathcal{B}$ responds with the challenge ciphertext $(\boldsymbol{c}, d)$.
- $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

By Corollary 2.10, $\mathcal{B}$ runs in polynomial time. Moreover, the probability that $\mathcal{B}$ succeeds is exactly the probability that $\mathcal{A}$ succeeds, as needed. (In particular, notice that the responses that $\mathcal{A}$ receives to all of its queries are distributed exactly as they are in $\mathbf{Game}_3$, including $\boldsymbol{t}$, which it receives in response to a random oracle query on id^*. The same holds for the master public key $\boldsymbol{P}$ and the challenge ciphertext $(\boldsymbol{c}, d)$.) □

Combining the claims together with security of the scheme in Section 3 presented in Theorem 3.2 yields the result. In particular, by applying the claims in sequence, we see that if there exists a polynomial-time adversary with non-negligible advantage in the full security game against the IBE scheme (i.e., $\mathbf{Game}_1$) then there exists another polynomial-time adversary with non-negligible advantage in the semantic security game against the scheme from Section 3. We can then apply Theorem 3.2 (in particular using the fact that $r \geq 2q^{1/k}\eta_\varepsilon^{(k)}(\boldsymbol{Q}')$) to reach a contradiction. □

4.3 Putting everything together

The following theorem follows immediately from combining Theorems 4.1 and 4.2.

Theorem 4.6. *Suppose that the parameters in the IBE scheme satisfy that* $s := s(n) \geq \sqrt{\log n}\|\boldsymbol{Q}\|$, $r := r(n) \geq \sqrt{n \log n}s/q$, $q = q(n) \geq 2^{n/2}$ *is prime,* $\rho = \rho(n) > \omega(\sqrt{\|\boldsymbol{Q}^{-1}\| \cdot n \log n})$, *and* $r\rho < o(1/\sqrt{\log n})$. *And, suppose that there exists some* $\boldsymbol{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$ *such that* $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ *is hard and* $r \geq 2q^{1/k} \cdot \eta_\varepsilon^{(k)}(\boldsymbol{Q}')$ *for some* $\varepsilon < \mathrm{negl}(n)$ *and* $k > \omega(\log n)$.

Then, $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a correct and fully secure identity-based encryption scheme.*

4.4 Instantiating our IBE scheme with specific quadratic forms and parameters

Thus far, we have described our IBE scheme abstractly in terms of an arbitrary quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and a rather large number of parameters with a number of relations between them (as in Theorem 4.6), and we proved security assuming the existence of some other quadratic form $\boldsymbol{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$ with sufficiently small $\eta^{(k)}$ for which $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard.

It remains to show that we can pick parameters and $\boldsymbol{Q}$ and $\boldsymbol{Q}'$ satisfying these constraints, and for which $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is plausibly hard.

To that end, we use the simple and elegant forms suggested by [DvW22] in a similar context. Specifically, we take

$$\boldsymbol{Q} = \begin{pmatrix} g\boldsymbol{I}_{n/2} & 0 \\ 0 & (g+1)\boldsymbol{I}_{n/2} \end{pmatrix},$$

where $g := g(n)$ is a positive integer to be chosen later. (We have assumed that n is even for simplicity.) And, we take[19]

$$\boldsymbol{Q}' = \begin{pmatrix} g(g+1)\boldsymbol{I}_{n/2} & 0 \\ 0 & \boldsymbol{I}_{n/2} \end{pmatrix}.$$

Notice that $\boldsymbol{Q}$ is not very different from a rescaling of the identity matrix, while $\boldsymbol{Q}'$ is quite unbalanced. Formally, this is captured by the fact that $\eta_\varepsilon^{(n/2)}(\boldsymbol{Q}') = \eta_\varepsilon(\boldsymbol{I}_{n/2}) = \sqrt{\log(n/\varepsilon)/\pi} + o(1)$ (see Equation (1)), while $\eta_\varepsilon^{(n/2)}(\boldsymbol{Q}) = \eta_\varepsilon^{(n/2)}(g\boldsymbol{I}_{n/2}) = \sqrt{g}\eta_\varepsilon^{(n/2)}(\boldsymbol{Q}')$, which is quite a bit larger.

[19] These quadratic forms $\boldsymbol{Q}, \boldsymbol{Q}'$ are carefully chosen to have the same *genus*. For example, if we instead made the seemingly very similar choice $\boldsymbol{P} = g\boldsymbol{I}_n$ and $\boldsymbol{P}' = g^2\boldsymbol{I}_{n/2} \oplus \boldsymbol{I}_{n/2}$ instead of $\boldsymbol{Q}$ and $\boldsymbol{Q}'$, one might expect the scheme to behave similarly. Unfortunately, this is not the case, as $\mathcal{D}_s$-acLIP$^{\boldsymbol{P},\boldsymbol{P}'}$ is trivial to solve. Indeed, notice that for these two forms, any $\boldsymbol{S} \equiv \boldsymbol{P}$ must have all entries divisible by g, while any $\boldsymbol{S} \equiv \boldsymbol{P}'$ must have at least some entries not divisible by g. This phenomenon might worry the reader, as perhaps a similar attack could be used to solve $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$. But, this phenomenon is actually well understood in terms of the genus of a quadratic form. Roughly speaking, the genus captures any "arithmetic" invariant (like divisibility of all entries by g) of the quadratic form, and [DvW22] carefully chose $\boldsymbol{Q}$ and $\boldsymbol{Q}'$ so that they have the same genus and therefore "cannot be distinguished arithmetically." See [DvW22] for more discussion.

Now we discuss how to set the parameters r, s, q, ρ, and g. First, for security, we need $r \geq 2q^{2/n} \cdot \eta_\varepsilon^{(n/2)}(\boldsymbol{Q}')$ for some negligible ε. (This is used in the security argument when we argue that if $\boldsymbol{Q}$ is replaced by $\boldsymbol{Q}'$, then the resulting ciphertext is statistically close to random.) Therefore, from the above, we see that it suffices to take any $r > \omega(q^{2/n}\sqrt{\log n})$. We take $r = \log n$ for simplicity (since we do not care much about polylogarithmic factors, and since we will eventually take $q \leq 2^{O(n)}$). For correctness, we need that $r\rho < o(1/\sqrt{\log n})$, and we can therefore take any $\rho < o(1/\log^{3/2} n)$. So, we take $\rho = 1/\log^2 n$ for simplicity.

To set g, recall that we need that $\|\boldsymbol{Q}^{-1}\| < o(\rho^2/(n \log n)) = o(1/(n \log^5 n))$. (This is needed for security, particularly for Claim 4.4.) So, noting that $\|\boldsymbol{Q}^{-1}\| = 1/g$, we see that it suffices to take, say,

$$g = n \log^6 n \ .$$

It remains to set s and q, and here we have two natural choices. We need $s \geq \sqrt{\log(n)\|\boldsymbol{Q}\|} = \sqrt{\log(n)(g+1)} = \sqrt{\log(n)(n \log^6 n + 1)}$ (in order for $\mathcal{D}_s$ to work). So, we could simply take, say, $s = 2\sqrt{n}\log^{7/2}(n)$. We then anyway need $q \geq 2^{n/2}$ to be a prime for security, so we can simply take q to be some prime between, say, $2^{n/2}$ and $2^{n/2+1}$. The only remaining requirement is that $q \geq s\sqrt{n \log n}/r$, which is needed so that the sampling in the encryption algorithm can be done efficiently (using Theorem 2.18), which is trivially satisfied here (for sufficiently large n). Then, we obtain security under the assumption that $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard.

But, we might wish to invoke the worst-case hardness in Theorem 2.18. In order to do that, we need $s \geq 2^{n/2} \cdot \sqrt{g(g+1)}$, so we can take, say, $s = 2^{n/2+1}g = 2^{n/2+1} n \log^6 n$. And, we must have $q \geq s\sqrt{n \log n}/r = 2^{n/2+1} n^{3/2} \log^{5.5} n$, so we must take q to be slightly larger. This choice of parameters is slightly less efficient, mostly because the larger choice of s means that $\|\boldsymbol{P}\| \approx \mathrm{poly}(n)s^2$ will be larger, leading to a larger public key and slower operations involving $\boldsymbol{P}$. (See Theorem 2.18.) In exchange for this penalty in efficiency, we can invoke the worst-case to average-case reduction in Theorem 2.18 (which was proven by [DvW22]). And, we obtain security under the assumption that wcLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard.

All of the above parameter choices are quite similar to what one obtains for the IBE scheme in [GPV08] that our IBE scheme is based on. In particular, the best known attacks on LIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ (either the worst-case version or the average-case version) use essentially the same algorithms as the best known attacks on [GPV08], with nearly identical parameters. Specifically, the best known attacks on LIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ work via reduction to the problem of approximating the length of a shortest non-zero vector in a lattice up to a factor of $\sqrt{g}$. (And the best known attacks on [GPV08] are essentially the same.) For any polynomial approximation factor, our best known algorithms for this problem run in time $2^{\Theta(n)}$, and this is widely believed to be optimal. See [DvW22, Section 7] for more discussion (including discussion that one might heuristically expect to solve LIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ using a slightly faster (but still $2^{\Theta(n)}$-time) algorithm that in the worst case only approximates the length of the shortest non-zero lattice vector up to the slightly larger factor $\sqrt{ng}$).

5 A public-key encryption scheme that we will use to build FHE

In this section, we present a simple public-key encryption scheme that is very similar to the scheme presented in [BGPS23]. But, our new version has the advantage that we can modify it to make it fully homomorphic, which we do in Section 6.

Recall that the high-level idea behind the scheme in [BGPS23] is as follows. There is some fixed, "nice" quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$. Indeed, in [BGPS23], $\boldsymbol{Q}$ is simply the identity matrix, which is certainly a "nice" quadratic form. More generally, we will assume that $\boldsymbol{Q}$ is such that there exist some known linearly independent vectors $\boldsymbol{Z} := (\boldsymbol{z}_1, \ldots \boldsymbol{z}_k) \in \mathbb{Z}^{n \times k}$ such that $\|\boldsymbol{z}_i\|_{\boldsymbol{Q}^{-1}}^2 = \boldsymbol{z}_i^\top \boldsymbol{Q}^{-1} \boldsymbol{z}_i$ is rather small—notice that the norm is with respect to $\boldsymbol{Q}^{-1}$ and not $\boldsymbol{Q}$. (In terms of lattices, these are the coordinates of short linearly independent vectors in the *dual* basis.) Of course, if $\boldsymbol{Q}$ is the identity matrix, then one should naturally take the $\boldsymbol{Z}$ to be the identity matrix as well. [BGPS23] therefore did not describe the matrix $\boldsymbol{Z}$ explicitly. (More generally, our discussion below is really of our modified scheme, which is more general and does not require $\boldsymbol{Q}$ to be the identity matrix.)

The key-generation algorithm samples a unimodular transformation $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ (sampled appropriately), sets the public key to be the quadratic form $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}$, and the secret key is $\boldsymbol{S} := (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k) := \boldsymbol{U}^\top \boldsymbol{Z} \in \mathbb{Z}^{n \times k}$. So, $\boldsymbol{P}$ is a quadratic form that is equivalent to $\boldsymbol{Q}$, and notice that the $\boldsymbol{s}_i \in \mathbb{Z}^n$ have the property that

$$\|\boldsymbol{s}_i\|_{\boldsymbol{P}^{-1}}^2 = (\boldsymbol{U}^\top \boldsymbol{z}_i)^\top \boldsymbol{U}^{-1} \boldsymbol{Q}^{-1} \boldsymbol{U}^{-\top} \boldsymbol{U}^\top \boldsymbol{z}_i = \|\boldsymbol{z}_i\|_{\boldsymbol{Q}^{-1}}^2$$

is small.

The idea here is that $\boldsymbol{S}$ can be used to roughly determine whether for a given vector $\boldsymbol{t} \in \mathbb{R}^n$, $\mathrm{dist}_{\boldsymbol{P}}(\boldsymbol{t}, \mathbb{Z}^n)$ is large or small. In particular, notice that if $\boldsymbol{y} \in \mathbb{Z}^n$ such that $\|\boldsymbol{t} - \boldsymbol{y}\|_{\boldsymbol{P}}$ is small, then

$$\langle \boldsymbol{t}, \boldsymbol{s}_i \rangle = \langle \boldsymbol{t} - \boldsymbol{y}, \boldsymbol{s}_i \rangle \bmod 1 \ .$$

And, by the Cauchy-Schwarz inequality (Equation (2)), we have that

$$|\langle \boldsymbol{t} - \boldsymbol{y}, \boldsymbol{s}_i \rangle| \leq \|\boldsymbol{t} - \boldsymbol{y}\|_{\boldsymbol{P}} \|\boldsymbol{s}_i\|_{\boldsymbol{P}^{-1}} = \|\boldsymbol{t} - \boldsymbol{y}\|_{\boldsymbol{P}} \|\boldsymbol{z}_i\|_{\boldsymbol{Q}^{-1}} \ ,$$

where the terms on the right-hand side are small by assumption. So, we have that if $\mathrm{dist}_{\boldsymbol{P}}(\boldsymbol{t}, \mathbb{Z}^n) \ll 1/\|\boldsymbol{s}_i\|_{\boldsymbol{P}^{-1}}$ then $\langle \boldsymbol{t}, \boldsymbol{s}_i \rangle$ will be close to an integer (i.e., small modulo 1).

With this, we can describe our decryption procedure (admittedly before we have explained how to encrypt). We do not wish to work with real numbers modulo 1, so we instead work with $\mathbb{Z}_q$ where q is some discretization parameter. ([BGPS23] does not bother with this discretization. But, for us, it will matter when we build FHE. We could work with $\mathbb{T}_q$ instead of $\mathbb{Z}_q$ as we did in Sections 3 and 4, but this choice yields significantly nicer notation.) Ciphertexts will be $\boldsymbol{c} \in \mathbb{Z}_q^n$, and to decrypt, we compute

$$d := \|\boldsymbol{S}^\top \boldsymbol{c} \bmod q\|^2 = \sum_{i=1}^{k} (\langle \boldsymbol{c}, \boldsymbol{s}_i \rangle \bmod q)^2 \ ,$$

where we represent values modulo q according to their representatives in the range $[-q/2, q/2)$. If d is smaller than some threshold, we output 0. Otherwise, we output 1. In the case when $\boldsymbol{Q}$ is the identity matrix considered in [BGPS23], this sum is precisely equal to $\mathrm{dist}_{\boldsymbol{P}}(\boldsymbol{c}, q\mathbb{Z}^n)^2$. Here, we think of this sum as a sort of proxy for this distance. (Indeed, in [BGPS23] they can simply assume that such distances can be computed directly, which is true when $\boldsymbol{Q} = \boldsymbol{I}_n$. They therefore do not explicitly use matrices $\boldsymbol{Z}$ and $\boldsymbol{S}$. We are careful to be more general.)

We next describe how to encrypt a bit $\mu \in \{0, 1\}$ into a ciphertext $\boldsymbol{c}_\mu$ under the public key $\boldsymbol{P}$. Naturally, we want our ciphertexts to have the property that encryptions $\boldsymbol{c}_0 \in \mathbb{Z}_q^n$ of 0 should have small inner product with all of the $\boldsymbol{s}_i$. And, we have already seen that it suffices to take $\boldsymbol{c}_0$ such that

$\mathrm{dist}_{\boldsymbol{P}}(\boldsymbol{c}_0, q\mathbb{Z}^n)$ is small. To that end, we sample $\boldsymbol{x} \in \mathbb{Z}^n$ (from an appropriate distribution—namely, a discrete Gaussian with variance proportional to $\boldsymbol{P}$) so that $\|\boldsymbol{x}\|_{\boldsymbol{P}} \ll q/\|\boldsymbol{s}_i\|_{\boldsymbol{P}^{-1}}$ and then set $\boldsymbol{c}_0 := \boldsymbol{x} \bmod q$. The analysis above immediately tells us that $\boldsymbol{c}_0$ will decrypt to 0.

To encrypt $\mu = 1$ under the public key $\boldsymbol{P}$, one can simply sample a ciphertext $\boldsymbol{c}_1 \in \mathbb{Z}_q^n$ that is uniformly random as is done in [BGPS23]. However, we will need a bit more structure for our eventual FHE scheme, so we instead set $\boldsymbol{c}_1 := \boldsymbol{x} + \boldsymbol{h} \bmod q$ where $\boldsymbol{x}$ is sampled as above (from a discrete Gaussian) and $\boldsymbol{h} \in \mathbb{Z}_q^n$ is an additional element included in the public key with the property that $\|\boldsymbol{S}^\top \boldsymbol{h} \bmod q\|^2$ is large.

This way of doing things yields a pleasant symmetry, as now our ciphertext $\boldsymbol{c}_\mu$ has the form

$$\boldsymbol{c}_\mu = \boldsymbol{x} + \mu \boldsymbol{h} \bmod q\ .$$

Of course, readers who have seen Regev's encryption scheme [Reg05] (or its many variants) will recognize this as being remarkably similar, which is certainly not a coincidence. From our perspective, it is crucial for our ciphertexts to have this form (specifically, to have this form with some $\boldsymbol{h}$ that is fixed by the public key) in order to make our scheme compatible with the gadget-matrix-based techniques that we will use in Section 6 to build FHE.

However, there is one subtlety left to worry about. Where does $\boldsymbol{h} \in \mathbb{Z}_q^n$ come from?[20] It would be relatively natural to choose some fixed $\boldsymbol{h}' \in \mathbb{Z}_q^n$ such that $\|\boldsymbol{Z}^\top \boldsymbol{h}' \bmod q\|^2$ is large and to set $\boldsymbol{h} := \boldsymbol{U}^{-1}\boldsymbol{h}'$. (E.g., if $\boldsymbol{Z} = \boldsymbol{I}_n$ is simply the identity, then it is natural to set $\boldsymbol{h}' = (q/2, \dots, q/2)^\top$.) But, it is not clear how to prove that the scheme remains secure if we include such an $\boldsymbol{h}$ in the public key, since such an $\boldsymbol{h}$ depends on the secret value $\boldsymbol{U}$. Instead, we simply set $\boldsymbol{h} \sim \mathbb{Z}_q^n$ to be uniformly random, which of course leaks nothing about $\boldsymbol{U}$. We then observe that as long as the $\boldsymbol{s}_i$ are linearly independent modulo q (which is equivalent to the $\boldsymbol{z}_i$ being linearly independent modulo q), it is easy to see that the inner products $\langle \boldsymbol{s}_i, \boldsymbol{h} \rangle \bmod q \in \mathbb{Z}_q$ are uniformly random and independent. Therefore,

$$\|\boldsymbol{S}^\top \boldsymbol{h} \bmod q\|^2 = \sum_{i=1}^{k} (\langle \boldsymbol{s}_i, \boldsymbol{h} \rangle \bmod q)^2$$

will be the sum of k independent random variables, so it will be highly concentrated around its rather large mean (which when q is even is precisely equal to $k(q^2+2)/12$).

Notice that, in order to achieve this concentration, we need k to be reasonably large. (In fact, $k > \omega(\log n)$ is sufficient for negligible error.) This in particular is why we do not simply take $k = 1$. (One can view Regev encryption as a variant of this scheme in which $k = 1$ but in which the public key is sampled in a very different way that allows one to take $\boldsymbol{h} = (q/2)\boldsymbol{e}_{n+1}$.)

5.1 The scheme

Our public-key encryption scheme comes with public parameters consisting of a quadratic form $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, a parameter $s := s(n) \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}$ (which roughly controls the size of $\boldsymbol{P}$), a noise parameter $r := r(n) \geq \sqrt{n \log n}\, s/q$, and a discretization parameter $q := q(n) \geq 10$.[21]

[20] In Regev's encryption scheme, $\boldsymbol{h}$ is effectively $(q/2)\boldsymbol{e}_{n+1}$, though the scheme is typically not written in this way.

[21] In Sections 3 and 4, we needed q to be prime, but we note that q does *not* need to be prime here. Indeed, in Section 6, we will use this scheme with q a power of two because this makes bit decomposition of elements in $\mathbb{Z}_q$ particularly simple. At a technical level, we needed q to be prime in Sections 3 and 4 because in those sections we could only guarantee that $\eta_\varepsilon^{(k)}(\boldsymbol{Q}')$ was small, and we then applied Theorem 2.8 to show that the inner product

We also require that there exist known vectors $\boldsymbol{Z} := (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k) \in \mathbb{Z}^{n \times k}$ that are linearly independent modulo q and such that $\|\boldsymbol{z}_i\|_{\boldsymbol{Q}^{-1}}$ is small.

- $\mathsf{Gen}(1^n)$: Sample $\boldsymbol{U} \leftarrow \mathcal{D}_s(\boldsymbol{Q}) \in \mathrm{GL}_n(\mathbb{Z})$, and compute $\boldsymbol{P} = \boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}$ (as in Theorem 2.18). Sample $\boldsymbol{h} \sim \mathbb{Z}_q^n$. Set $\mathsf{sk} := \boldsymbol{S} := \boldsymbol{U}^\top \boldsymbol{Z} \in \mathbb{Z}^{n \times k}$ and $\mathsf{pk} := (\boldsymbol{P}, \boldsymbol{h})$.
- $\mathsf{Enc}(\mathsf{pk}, \mu \in \{0,1\})$: Sample $\boldsymbol{x} \sim D_{\boldsymbol{P}, rq}$ (using the $\mathcal{A}$ algorithm from Theorem 2.18). Output $\boldsymbol{c} := \boldsymbol{x} + \mu \boldsymbol{h} \bmod q \in \mathbb{Z}_q^n$.
- $\mathsf{Dec}(\mathsf{sk}, \boldsymbol{c})$: Compute
$$d := \|\boldsymbol{S}^\top \boldsymbol{c} \bmod q\|^2 ,$$
where we represent elements modulo q in the interval $[-q/2, q/2)$. Output 0 if $d < kq^2/100$ and 1 otherwise.

Notice in particular that the condition that $s \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}$ is needed for the key-generation algorithm to run $\mathcal{D}_s$ (see Theorem 2.18), while the fact that $qr \geq s\sqrt{n \log n}$ implies that the algorithm in Theorem 2.18 used by the encryption algorithm correctly generates a sample $\boldsymbol{x} \sim D_{\boldsymbol{P}, rq}$ in polynomial time.

5.2 Correctness

Theorem 5.1. *Suppose that the parameters $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, $r := r(n) \geq \sqrt{n \log n} \cdot s/q$, an integer $q = q(n) \geq 10$, and $\boldsymbol{Z} := (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k) \in \mathbb{Z}^{n \times k}$ in the encryption scheme above satisfy*

$$r \leq \frac{1}{10\sqrt{n d_z}} ,$$

where $d_z := \frac{1}{k} \cdot \sum_{i=1}^k \|\boldsymbol{z}_i\|_{\boldsymbol{Q}^{-1}}^2$.

Then, the above scheme is correct, in the sense that for any $\mu \in \{0,1\}$,

$$\Pr_{(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}(1^n)}[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mu)) = \mu] \geq 1 - 2^{-\Omega(k)} - 2^{-n} .$$

More generally, with probability at least $1 - 2^{-\Omega(k)}$ over the choice of $\boldsymbol{h} \sim \mathbb{Z}_q^n$, we have

$$\mathsf{Dec}(\mathsf{sk}, \boldsymbol{x} + \mu \boldsymbol{h} \bmod q) = \mu$$

for any $\boldsymbol{x} \in \mathbb{Z}^n$ with $\|\boldsymbol{x}\|_{\boldsymbol{P}} < \frac{q}{10\sqrt{d_z}}$.

Proof. It suffices to prove the "more generally" statement. The correctness of the encryption algorithm then follows from Theorem 2.13, which tells us that $\Pr_{\boldsymbol{x} \sim D_{\boldsymbol{P}, qr}}[\|\boldsymbol{x}\|_{\boldsymbol{P}} \geq \sqrt{n} qr] < 2^{-n}$.

To prove the "more generally", let

$$d := \|\boldsymbol{S}^\top \boldsymbol{c} \bmod q\|^2 = \sum_{i=1}^k (\langle \boldsymbol{s}_i, \boldsymbol{x} \rangle + \mu \langle \boldsymbol{s}_i, \boldsymbol{h} \rangle \bmod q)^2 ,$$

$\langle \boldsymbol{e} - \boldsymbol{c}, \boldsymbol{t} \rangle$ is close to uniform (when the public key is replaced by $\boldsymbol{Q}'$). In this section (and in Section 6), we will be able to take $\eta_\varepsilon(\boldsymbol{Q}')$ itself to be small (with no k), which means that $\boldsymbol{c}$ itself is uniformly random (when the public key is replaced by $\boldsymbol{Q}'$), with no need for an inner product and no need for primality of q.

as in the decryption algorithm. Let

$$d' := \|\boldsymbol{S}^\top \boldsymbol{x}\|^2 = \sum_{i=1}^{k} \langle \boldsymbol{s}_i, \boldsymbol{x} \rangle^2 \ .$$

It follows from the Cauchy-Schwarz inequality (Equation (2)) that

$$d' \leq \|\boldsymbol{x}\|_{\boldsymbol{P}}^2 \sum_{i=1}^{k} \|\boldsymbol{s}_i\|_{\boldsymbol{P}^{-1}}^2 < \frac{kq^2}{100} \ ,$$

where we have used the fact that $\sum_{i=1}^{k} \|\boldsymbol{s}_i\|_{\boldsymbol{P}^{-1}}^2 = \sum_{i=1}^{k} \|\boldsymbol{z}_i\|_{\boldsymbol{Q}^{-1}}^2 = kd_z$. This immediately implies that the decryption algorithm outputs 0 when $\mu = 0$, as needed.

When $\mu = 1$, we need to also consider

$$d_{\boldsymbol{h}} := \|\boldsymbol{S}^\top \boldsymbol{h} \bmod q\|^2 = \sum_{i=1}^{k} (\langle \boldsymbol{s}_i, \boldsymbol{h} \rangle \bmod q)^2 \ .$$

Indeed, we have from triangle inequality that

$$\sqrt{d} \geq \sqrt{d_{\boldsymbol{h}}} - \sqrt{d'} \geq \sqrt{d_{\boldsymbol{h}}} - \sqrt{k}q/10 \ .$$

So, it suffices to prove that $d_{\boldsymbol{h}} \geq kq^2/25$ with high probability over $\boldsymbol{h} \sim \mathbb{Z}_q^n$.

To that end, notice that because $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ are linearly independent modulo q and $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ has determinant ± 1, $\boldsymbol{s}_1 := \boldsymbol{U}^\top \boldsymbol{z}_1, \ldots, \boldsymbol{s}_k := \boldsymbol{U}^\top \boldsymbol{z}_k$ are also linearly independent modulo q. It follows that the random variables $\langle \boldsymbol{s}_i, \boldsymbol{h} \rangle \in \mathbb{Z}_q$ are uniformly random and independent. So, we can write $d_{\boldsymbol{h}} = \sum_{i=1}^{k} \chi_i^2$ where $\chi_i \sim [-q/2, q/2)$ are independent and uniform. A simple computation then shows that $\mathbb{E}[d_{\boldsymbol{h}}] = k(q^2+2)/12$ if q is even and $\mathbb{E}[d_{\boldsymbol{h}}] = k(q^2-1)/12 > kq^2/15$ if q is odd. Either way, by the Chernoff-Hoeffding bound, we have that $d_{\boldsymbol{h}} > kq^2/25$ except with probability at most $2^{-\Omega(k)}$, as needed. □

5.3 Security

Theorem 5.2. *Suppose that there exists some $\boldsymbol{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$ such that for the parameters $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, $s := s(n) \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}$, and $r := r(n) \geq \sqrt{n \log n} \cdot s/q$ in the encryption scheme above, we have that $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard and $r \geq \eta_\varepsilon(\boldsymbol{Q}')$ for some $\varepsilon < \mathrm{negl}(n)$. Then, the ciphertext $\boldsymbol{c} \in \mathbb{Z}_q^n$ produced by the encryption algorithm is computationally indistinguishable from uniformly random $\boldsymbol{u} \sim \mathbb{Z}_q^n$.*

Proof. Consider $\mathbf{Game}_1$ and $\mathbf{Game}_2$, where in $\mathbf{Game}_1$ the goal is to distinguish $\boldsymbol{c} = \boldsymbol{x} + \mu \boldsymbol{h} \bmod q$ for $\boldsymbol{x} \sim D_{\boldsymbol{P},rq}$ from a uniformly random element $\boldsymbol{u} \sim \mathbb{Z}_q^n$. $\mathbf{Game}_2$ is the same as $\mathbf{Game}_1$, but with $\boldsymbol{U}$ drawn from $\mathcal{D}_s(\boldsymbol{Q}')$ instead of $\mathcal{D}_s(\boldsymbol{Q})$ and the public key set to $\boldsymbol{P}' := \boldsymbol{U}^\top \boldsymbol{Q}' \boldsymbol{U}$ instead of $\boldsymbol{P} := \boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}$. By the hardness of $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$, the advantage of any efficient adversary in $\mathbf{Game}_1$ is at most negligibly larger than its advantage in $\mathbf{Game}_2$. Now we argue that *any* adversary (even computationally unbounded adversaries) in $\mathbf{Game}_2$ has advantage at most 2ε.

Indeed, since $\boldsymbol{x} \sim D_{\boldsymbol{P}',rq}$ and $r \geq \eta_\varepsilon(\boldsymbol{Q}') = \eta_\varepsilon(\boldsymbol{P}')$ by assumption, we have that $\boldsymbol{x} \bmod q$ is within statistical distance $2\varepsilon < \mathrm{negl}(n)$ of uniformly random by Lemma 2.11. The result follows. □

6 Fully homomorphic encryption

We next present our FHE scheme.

6.1 The encryption scheme

Our FHE scheme uses the same set of parameters $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z}), s := s(n) \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}, r := r(n) \geq \sqrt{n \log n} s/q$, and $q := q(n) \geq 10$ from Section 5, as well as the set of known vectors $\boldsymbol{Z} := (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k) \in \mathbb{Z}^{n \times k}$ such that $\|\boldsymbol{z}_i\|_{\boldsymbol{Q}^{-1}}$ is small.

We will assume for simplicity that q is a power of 2. Let $m = n \log_2 q + 1$. As in [MP12, GSW13], we define the gadget *row* vector $\boldsymbol{g} := (1, 2, 4, \ldots, q/2) \in \mathbb{Z}_q^{\log_2 q}$, and the gadget matrix $\boldsymbol{G}$ to be

$$\boldsymbol{G} := \boldsymbol{I}_n \otimes \boldsymbol{g} = \begin{pmatrix} \boldsymbol{g} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{g} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{g} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{g} \end{pmatrix} \in \mathbb{Z}_q^{n \times (m-1)} . \tag{3}$$

Then, our encryption scheme is as follows.

- $\mathsf{Gen}(1^n)$: Sample $\boldsymbol{U} \leftarrow \mathcal{D}_s(\boldsymbol{Q}) \in \mathrm{GL}_n(\mathbb{Z})$, and compute $\boldsymbol{P} = \boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U}$. Sample $\boldsymbol{h} \sim \mathbb{Z}_q^n$. Set $\mathsf{sk} := \boldsymbol{S} := \boldsymbol{U}^\top \boldsymbol{Z} \in \mathbb{Z}^{n \times k}$ and $\mathsf{pk} := (\boldsymbol{P}, \boldsymbol{h})$.
- $\mathsf{Enc}(\mathsf{pk}, \mu \in \{0,1\})$: Sample $\boldsymbol{X} \leftarrow D_{\boldsymbol{P}, rq}^m$. Output $\boldsymbol{C} := \boldsymbol{X} + \mu \boldsymbol{H} \bmod q$, where
$$\boldsymbol{H} := (\boldsymbol{h}, \boldsymbol{G}) \in \mathbb{Z}_q^{n \times m} .$$
- $\mathsf{Dec}(\mathsf{sk}, \boldsymbol{C})$: Compute
$$d := \|\boldsymbol{S}^\top \boldsymbol{C}_1 \bmod q\|^2 ,$$
where $\boldsymbol{C}_1$ is the first column of $\boldsymbol{C}$ and we represent elements modulo q in the interval $[-q/2, q/2)$. Output 0 if $d < kq^2/100$ and 1 otherwise.

Notice that this scheme is very similar to the scheme in Section 5 except that our ciphertext $\boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$ is an entire matrix

$$\boldsymbol{C} = \boldsymbol{X} + \mu \boldsymbol{H} \bmod q ,$$

where $\boldsymbol{X} \sim D_{\boldsymbol{P}, rq}^m$ instead of just a single column vector $\boldsymbol{c} = \boldsymbol{x} + \mu \boldsymbol{h} \bmod q$ (where $\boldsymbol{x} \sim D_{\boldsymbol{P}, rq}$). Indeed, the first column $\boldsymbol{C}_1$ of $\boldsymbol{C}$ is precisely the same as the ciphertext $\boldsymbol{c}$ generated by the scheme in Section 5. The remaining columns can be viewed as independent encryptions of 0 plus a term that depends on μ (specifically, plus $\mu \boldsymbol{G}$). As we will see, this means that Theorem 5.2 more-or-less immediately implies the security of the above scheme. It also immediately implies that the above scheme is correct as a public-key encryption scheme, but we do not bother to show this, since we will prove the much stronger fact that there exists an Eval algorithm making the scheme fully homomorphic.

(The relationship between the two encryption schemes as we describe it above is nearly identical to the relationship between Regev's encryption scheme [Reg05] and the GSW fully homomorphic encryption scheme [GSW13]. Of course, this is not a coincidence, as we designed our scheme to mimic [GSW13].)

6.2 Security

Theorem 6.1. *Suppose that there exists some $\boldsymbol{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$ such that for the parameters $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$, $s := s(n) \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}$ and $r := r(n) \geq \sqrt{n \log n} \cdot s/q$ in the encryption scheme above, $\mathcal{D}_s$-$\mathrm{acLIP}^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard and $r \geq \eta_\varepsilon(\boldsymbol{Q}')$ for some $\varepsilon < \mathrm{negl}(n)$. Then, the ciphertext $\boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$ produced by the encryption algorithm is computationally indistinguishable from uniformly random $\boldsymbol{U} \sim \mathbb{Z}_q^{n \times m}$. In particular, the scheme is secure.*

Proof. Notice that each column $\boldsymbol{C}_i = \boldsymbol{X}_i + \mu \boldsymbol{H}_i \bmod q$ of the ciphertext is sampled independently, and that these columns can be written as $\boldsymbol{C}_i = \boldsymbol{c}_i + \mu \boldsymbol{H}_i \bmod q$, where $\boldsymbol{c}_i$ is distributed identically to an encryption of 0 produced by the scheme from Section 5. By Theorem 5.2, $\boldsymbol{c}_i$ is computationally indistinguishable from a random $\boldsymbol{u} \sim \mathbb{Z}_q^n$. Thus, by a hybrid argument, the whole ciphertext $\boldsymbol{C}$ is computationally indistinguishable from uniformly random $\boldsymbol{U} \sim \mathbb{Z}_q^{n \times m}$. □

6.3 The Eval function and homomorphic operations

Suppose that we wish to homomorphically evaluate a $(\oplus, \times)$-circuit $f : \{0,1\}^\ell \to \{0,1\}$ on input bits $\mu_1, \ldots, \mu_\ell \in \{0,1\}$. Notice that Theorem 5.1 immediately implies that our decryption algorithm will output μ whenever its input has the form $\boldsymbol{C} = \boldsymbol{X} + \mu \boldsymbol{H} \bmod q$ for $\boldsymbol{X}$ that is sufficiently small (in the $\boldsymbol{P}$ norm). So, it suffices to show how to convert ciphertexts $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_\ell$ of the form $\boldsymbol{C}_i = \boldsymbol{X}_i + \mu_i \boldsymbol{H} \bmod q$ into a ciphertext $\boldsymbol{C}_f = \boldsymbol{X}_f + \mu_f \boldsymbol{H} \bmod q$ of this form, where $\mu_f = f(\mu_1, \ldots, \mu_\ell)$.

Before defining how to perform homomorphic operations on ciphertexts, we define the (right) inverse map $I_{\boldsymbol{H}} : \mathbb{Z}_q^{n \times m} \to \{0,1\}^{m \times m}$ of $\boldsymbol{H}$ as

$$I_{\boldsymbol{H}}(\boldsymbol{C}) = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathrm{BitDecomp}(C_{1,1}) & \mathrm{BitDecomp}(C_{1,2}) & \cdots & \mathrm{BitDecomp}(C_{1,m}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{BitDecomp}(C_{n,1}) & \mathrm{BitDecomp}(C_{n,2}) & \cdots & \mathrm{BitDecomp}(C_{n,m}) \end{pmatrix} \in \{0,1\}^{m \times m} ,$$

where for $a \in \mathbb{Z}_q$, $a = \sum_{i=1}^{\log_2 q} x_i 2^{i-1}$ for bits $x_1, \ldots, x_{\log_2 q} \in \{0,1\}$ the bit decomposition function $\mathrm{BitDecomp} : \mathbb{Z}_q \to \{0,1\}^{\log_2 q}$ is defined as

$$\mathrm{BitDecomp}(a) = \mathrm{BitDecomp}\left(\sum_{i=1}^{\log_2 q} x_i 2^{i-1} \right) = (x_1, x_2, \ldots, x_{\log_2 q})^\top \in \{0,1\}^{\log_2 q} .$$

Notice that $I_{\boldsymbol{H}}(\boldsymbol{C})$ is a matrix with small entries (in fact $\{0,1\}$ entries), and it is actually an inverse of $\boldsymbol{H}$ in the sense that

$$\boldsymbol{H} I_{\boldsymbol{H}}(\boldsymbol{C}) = \boldsymbol{C} .$$

(In particular, we have that $\boldsymbol{g} \cdot \mathrm{BitDecomp}(c) = c$ for any $c \in \mathbb{Z}_q$.) This is exactly the same as the standard inverse map used with the gadget matrix $\boldsymbol{G}$ in, e.g., [GSW13], except that our gadget matrix $\boldsymbol{H}$ has an extra column $\boldsymbol{h}$ prepended to it, and therefore $I_{\boldsymbol{H}}(\boldsymbol{C})$ has an extra row of zeros prepended to it.[22]

[22] There is different notation for maps of this form. Most recent works use the notation $\boldsymbol{H}^{-1}(\boldsymbol{C})$—or, rather, they write $\boldsymbol{G}^{-1}$ for the analogous map for $\boldsymbol{G}$. We prefer the notation $I_{\boldsymbol{H}}$ because we do not wish to suggest that $I_{\boldsymbol{H}}$ is a linear map. [GSW13] called their analogous map Flatten. These are all more-or-less the same thing.

Now, assume we have two ciphertexts

$$\boldsymbol{C}_1 = \boldsymbol{X}_1 + \mu_1 \boldsymbol{H} \bmod q \,, \qquad \boldsymbol{C}_2 = \boldsymbol{X}_2 + \mu_2 \boldsymbol{H} \bmod q \,,$$

where $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are in some sense "short." We would like to compute

$$\boldsymbol{C}_\oplus = \boldsymbol{X}_\oplus + (\mu_1 \oplus \mu_2)\boldsymbol{H} \bmod q \,, \qquad \boldsymbol{C}_\times = \boldsymbol{X}_\times + (\mu_1 \mu_2)\boldsymbol{H} \bmod q \,,$$

where $\boldsymbol{X}_\oplus$ and $\boldsymbol{X}_\times$ are also short. To that end, we define

$$\boldsymbol{C}_\times := \boldsymbol{C}_1 \cdot I_{\boldsymbol{H}}(\boldsymbol{C}_2) \bmod q \,,$$

and

$$\boldsymbol{C}_\oplus := \boldsymbol{C}_1 + \boldsymbol{C}_2 - 2\boldsymbol{C}_\times \bmod q \,,$$

which correspond to homomorphic evaluation of a $\times$ gate and a $\oplus$ gate, respectively. We will prove this carefully soon, but notice that by applying the identity $\boldsymbol{H} I_{\boldsymbol{H}}(\boldsymbol{C}) = \boldsymbol{C}$ and recalling that $I_{\boldsymbol{H}}(\boldsymbol{C})$ has small entries, $\boldsymbol{C}_\oplus$ and $\boldsymbol{C}_\times$ do in fact have the desired form. Indeed, applying this identity yields

$$\boldsymbol{C}_1 \cdot \boldsymbol{I}_{\boldsymbol{H}}(\boldsymbol{C}_2) = (\boldsymbol{X}_1 + \mu_1 \boldsymbol{H}) \cdot \boldsymbol{I}_{\boldsymbol{H}}(\boldsymbol{C}_2) = \boldsymbol{X}_1 \boldsymbol{I}_{\boldsymbol{H}}(\boldsymbol{C}_2) + \mu_1 \boldsymbol{C}_2 = \boldsymbol{X}_1 \boldsymbol{I}_{\boldsymbol{H}}(\boldsymbol{C}_2) + \mu_1 \boldsymbol{X}_2 + \mu_1 \mu_2 \boldsymbol{H} \bmod q$$

which has the desired form with $\boldsymbol{X}_\times := \boldsymbol{X}_1 \boldsymbol{I}_{\boldsymbol{H}}(\boldsymbol{C}_2) + \mu_1 \boldsymbol{X}_2$. And, $\boldsymbol{C}_\oplus$ works out similarly after applying the similar identity $\mu_1 \oplus \mu_2 = \mu_1 + \mu_2 - 2\mu_1\mu_2$. (Again, all of this is very closely analogous to [GSW13].)

Iterating the above construction yields the following definition and the Eval function for our scheme.

Definition 6.2. *For a $(\oplus, \times)$-circuit $f : \{0,1\}^\ell \to \{0,1\}$ of depth $L = L(n) \geq 0$ and ciphertexts $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_\ell \in \mathbb{Z}_q^{n \times m}$, we define $\boldsymbol{C}_f \in \mathbb{Z}_q^{n \times m}$ recursively as follows.*

1. *If $f(\mu_1, \ldots, \mu_\ell) = \mu_i$ for some $i \in [\ell]$ (i.e., if $L = 0$), $\boldsymbol{C}_f := \boldsymbol{C}_i$.*
2. *If $f(\mu_1, \ldots, \mu_\ell) = f_1(\mu_1, \ldots, \mu_\ell) \times f_2(\mu_1, \ldots, \mu_\ell)$ for some $(\oplus, \times)$-circuits f_1 and f_2 of depth at most $L - 1$, then $\boldsymbol{C}_f := \boldsymbol{C}_{f_1} \cdot I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) \bmod q$.*
3. *If $f(\mu_1, \ldots, \mu_\ell) = f_1(\mu_1, \ldots, \mu_\ell) \oplus f_2(\mu_1, \ldots, \mu_\ell)$ for some $(\oplus, \times)$-circuits f_1 and f_2 of depth at most $L - 1$, then $\boldsymbol{C}_f := \boldsymbol{C}_{f_1} + \boldsymbol{C}_{f_2} - 2\boldsymbol{C}_{f_1} \cdot I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) \bmod q$.*

Then, we define the evaluation algorithm as $\mathsf{Eval}(\mathsf{pk}, f, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_\ell) := \boldsymbol{C}_f$.

We next prove correctness of our fully homomorphic encryption scheme, which requires showing that $\mathsf{Eval}(\mathsf{pk}, f, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_\ell)$ outputs a ciphertext $\boldsymbol{C}_f = \boldsymbol{X}_f + \mu_f \boldsymbol{H}$ where $\|(\boldsymbol{X}_f)_i\|_P$ is relatively small for each of the columns of $(\boldsymbol{X}_f)_i$ of $\boldsymbol{X}_f$.

Theorem 6.3. *For a $(\oplus, \times)$-circuit $f : \{0,1\}^\ell \to \{0,1\}$ of depth L, if for all $1 \leq i \leq \ell$, $\boldsymbol{C}_i \leftarrow \mathsf{Enc}(pk, \mu_i)$, then $\boldsymbol{C}_f = \boldsymbol{X}_f + \mu_f \boldsymbol{H} \bmod q$, where $\boldsymbol{X}_f \in \mathbb{Z}^{n \times m}$ has columns $(\boldsymbol{X}_f)_i$ satisfying*

$$\|(\boldsymbol{X}_f)_i\|_{\boldsymbol{P}} \leq (2m+2)^L \sqrt{n} r q$$

with probability at least $1 - \ell m 2^{-n}$, where $\mu_f := f(\mu_1, \ldots, \mu_\ell)$.

In particular, so long as the parameter $r := r(n) \geq \sqrt{n \log n} \cdot s/q$ *in the encryption scheme is taken so that*

$$r \leq \frac{1}{10\sqrt{nd_z}(2m+2)^L},$$

where $d_z := \frac{1}{k}\sum_{i=1}^{k} \|\boldsymbol{z}_i\|^2_{\boldsymbol{Q}^{-1}}$*, we have*

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_\ell)) = f(\mu_1, \ldots, \mu_\ell)] \geq 1 - \ell m 2^{-n} - 2^{-\Omega(k)}.$$

In other words, our scheme is a level L *fully homomorphic encryption scheme for such parameters, provided that* $k > \omega(\log n)$*.*

Proof. First, let $\boldsymbol{C}_i := \boldsymbol{X}_i + \mu_i \boldsymbol{H} \bmod q$ where $\boldsymbol{X}_i \sim D^m_{\boldsymbol{P},rq}$ as in our encryption scheme. By Theorem 2.13 and the union bound, we have that for each i and j, the jth column $(\boldsymbol{X}_i)_j$ of $\boldsymbol{X}_i$ has $\|(\boldsymbol{X}_i)_j\|_{\boldsymbol{P}} < \sqrt{n}rq$ except with probability at most $\ell m 2^{-n}$.

Assuming that this holds, we prove the result by induction on the depth L of f. In the base case, if f has depth 0, then $\boldsymbol{X}_f = \boldsymbol{X}_i$ for some i, and we are done.

Otherwise, the circuit has depth $L \geq 1$, and there are two cases to consider, depending on whether $f = f_1 \times f_2$ or $f = f_1 \oplus f_2$ for circuits f_1 and f_2 with depth at most $L-1$.

In the first case, $f = f_1 \times f_2$ and $\boldsymbol{C}_f = \boldsymbol{C}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2})$. By induction, we can write $\boldsymbol{C}_{f_i} = \boldsymbol{X}_{f_i} + \mu_{f_i}\boldsymbol{H} \bmod q$ for $i \in \{1,2\}$, where $\|(\boldsymbol{X}_{f_i})_j\|_{\boldsymbol{P}} \leq (2m+2)^{L-1}\sqrt{n}rq$. Using the identity $\boldsymbol{H} I_{\boldsymbol{H}}(\boldsymbol{C}) = \boldsymbol{C}$, we see that

$$\begin{aligned}
\boldsymbol{C}_f &= (\boldsymbol{X}_{f_1} + \mu_{f_1}\boldsymbol{H}) I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) \bmod q \\
&= \boldsymbol{X}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) + \mu_{f_1}\boldsymbol{X}_{f_2} + \mu_{f_1}\mu_{f_2}\boldsymbol{H} \bmod q \\
&= \boldsymbol{X}_f + \mu_f \boldsymbol{H} \bmod q\ ,
\end{aligned}$$

where $\boldsymbol{X}_f := \boldsymbol{X}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) + \mu_{f_1}\boldsymbol{X}_{f_2}$ and $\mu_f := \mu_{f_1}\mu_{f_2}$.

Since $I_{\boldsymbol{H}}(\boldsymbol{C}_{\boldsymbol{f_2}}) \in \{0,1\}^{m \times m}$ it follows from the induction hypothesis that for $i = 1, \ldots, m$, the ith column of $\boldsymbol{X}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2})$ satisfies

$$\|(\boldsymbol{X}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}))_i\|_{\boldsymbol{P}} \leq \sum_{j=1}^{m} \|(\boldsymbol{X}_{f_1})_j\|_{\boldsymbol{P}} \leq m(2m+2)^{L-1}\sqrt{n}rq\ .$$

Furthermore, $\|\mu_{f_1}\boldsymbol{X}_{f_2}\|_{\boldsymbol{P}} \leq (2m+2)^{L-1}\sqrt{n}rq$ by the induction hypothesis, and so by triangle inequality we have $\|(\boldsymbol{X}_f)_i\|_{\boldsymbol{P}} \leq (2m+2)^L\sqrt{n}rq$, as needed.

The second case where $f = f_1 \oplus f_2$ is very similar. In that case, we have

$$\begin{aligned}
\boldsymbol{C}_f &= \boldsymbol{C}_{f_1} + \boldsymbol{C}_{f_2} - 2\boldsymbol{C}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) \bmod q \\
&= (\boldsymbol{X}_{f_1} + \mu_{f_1}\boldsymbol{H}) + (\boldsymbol{X}_{f_2} + \mu_{f_2}\boldsymbol{H}) - 2(\boldsymbol{X}_{f_1} + \mu_{f_1}\boldsymbol{H}) I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) \bmod q \\
&= \boldsymbol{X}_{f_1} - 2\boldsymbol{X}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) + \boldsymbol{X}_{f_2} - 2\mu_{f_1}\boldsymbol{X}_{f_2} + (\mu_{f_1} + \mu_{f_2} - 2\mu_{f_1}\mu_{f_2})\boldsymbol{H} \bmod q \\
&= \boldsymbol{X}_{f_1} - 2\boldsymbol{X}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) + (1 - 2\mu_{f_1})\boldsymbol{X}_{f_2} + (\mu_{f_1} \oplus \mu_{f_2})\boldsymbol{H} \bmod q \\
&= \boldsymbol{X}_f + \mu_f \boldsymbol{H} \bmod q\ ,
\end{aligned}$$

where $\boldsymbol{X}_f := \boldsymbol{X}_{f_1} - 2\boldsymbol{X}_{f_1} I_{\boldsymbol{H}}(\boldsymbol{C}_{f_2}) + (1 - 2\mu_{f_1})\boldsymbol{X}_{f_2}$ and $\mu_f := \mu_{f_1} \oplus \mu_{f_2}$. It again follows from the induction hypothesis and triangle inequality that $\|(\boldsymbol{X}_f)_i\|_{\boldsymbol{P}} \leq (2m+2)^L\sqrt{n}rq$, as needed.

Finally, since

$$\|(\boldsymbol{X}_f)_1\|_{\boldsymbol{P}}^2 \leq \frac{q^2}{100 d_z},$$

correctness follows by Theorem 5.1. □

Remark 6.4. *We note that our scheme inherits from [GSW13] a peculiar asymmetry in noise growth. In particular, when one homomorphically computes a ciphertext* $\boldsymbol{C}_f := \boldsymbol{X}_f + \mu_f \boldsymbol{H} \bmod q$ *from two ciphertexts* $\boldsymbol{C}_{f_1} := \boldsymbol{X}_{f_1} + \mu_{f_1}\boldsymbol{H} \bmod q$ *and* $\boldsymbol{C}_{f_2} := \boldsymbol{X}_{f_2} + \mu_{f_2}\boldsymbol{H} \bmod q$*, the resulting "noise"* $\boldsymbol{X}_f$ *satisfies* $\|\boldsymbol{X}_f\|_{\boldsymbol{P}} \leq (2m+1)\|\boldsymbol{X}_{f_1}\|_{\boldsymbol{P}} + \|\boldsymbol{X}_{f_2}\|_{\boldsymbol{P}}$*. In the above, we simply bound this by* $(2m+2)\max_i \|\boldsymbol{X}_{f_i}\|_{\boldsymbol{P}}$*. But, notice that we can potentially design our circuit* f *so that we know that* $\|\boldsymbol{X}_{f_1}\|_{\boldsymbol{P}}$ *is quite a bit smaller than* $\|\boldsymbol{X}_{f_2}\|_{\boldsymbol{P}}$*, in which case this bound is quite loose. This can allow us to evaluate much deeper circuits. For example, we can homomorphically evaluate the XOR of* ℓ *ciphertexts while only increasing the noise by a factor of* $\ell(2m+1)+1$*, while naively applying our theorem would suggest that the noise should increase by a factor of* $(2m+2)^{\log \ell}$ *(using a balanced circuit).*

In the case of [GSW13], this is sometimes quite useful. Specifically, this is useful for bootstrapping *(which we discuss in Section 6.6) when one is specifically interested in minimizing the noise growth in the decryption circuit. This was used in [BV14, AP14, DM15] to homomorphically evaluate the decryption circuit of (variants of) GSW encryption with noise growth bounded by a polynomial in* n*, even though it has depth* $\mathrm{poly}(\log n)$*. See Section 6.6 for discussion of bootstrapping in this context.*

6.4 Putting everything together

By combining the security and correctness results in Theorems 6.1 and 6.3, we obtain the following result.

Theorem 6.5. *Suppose that the parameters* $\boldsymbol{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$*,* $s := s(n) \geq \sqrt{\log(n)\|\boldsymbol{Q}\|}$*,* $r := r(n) \geq \sqrt{n \log n} s/q$*, and* $q := q(n) \geq 20$ *a power of two, and* $\boldsymbol{Z} := (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k) \in \mathbb{Z}^{n \times k}$ *in the homomorphic encryption scheme above satisfy that* $k > \omega(\log n)$ *and*

$$r < \frac{1}{20\sqrt{n d_z}},$$

where $d_z := \frac{1}{k} \cdot \sum_{i=1}^k \|\boldsymbol{z}_i\|_{\boldsymbol{Q}^{-1}}^2$ *and* $m := n \log_2 q + 1$*. Suppose further that there exists some* $\boldsymbol{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$ *such that* $\mathcal{D}_s$*-acLIP*$^{\boldsymbol{Q},\boldsymbol{Q}'}$ *is hard and* $r \geq \eta_\varepsilon(\boldsymbol{Q}')$ *for some* $\varepsilon < \mathrm{negl}(n)$*.*

Then, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is a correct and secure level-*L *fully homomorphic encryption scheme for any*

$$L \leq \frac{\log\left(\frac{1}{10\sqrt{n d_z} r}\right)}{\log(2m+2)}.$$

6.5 Instantiating our FHE scheme with specific quadratic forms and parameters

One possible instantiation of our FHE scheme is to use the quadratic form

$$\boldsymbol{Q} := \begin{pmatrix} g(g+1)\boldsymbol{I}_{n/2} & 0 \\ 0 & \boldsymbol{I}_{n/2} \end{pmatrix},$$

for some large positive integer g to be chosen later. This form was first suggested in a similar context in [DvW22]. The choice of vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ will just be $\boldsymbol{z}_1 := \boldsymbol{e}_1, \ldots, \boldsymbol{z}_{n/2} := \boldsymbol{e}_{n/2}$, with their $\boldsymbol{Q}^{-1}$ norms being

$$\|\boldsymbol{z}_i\|^2_{\boldsymbol{Q}^{-1}} = \frac{1}{g(g+1)} ,$$

so that $d_{\boldsymbol{z}} = \frac{1}{g(g+1)}$.

For security, again following [DvW22], we can consider

$$\boldsymbol{Q}' := \begin{pmatrix} g\boldsymbol{I}_{n/2} & 0 \\ 0 & (g+1)\boldsymbol{I}_{n/2} \end{pmatrix}.$$

Intuitively, this is a far more "balanced" quadratic form, and in particular it has smoothing parameter roughly proportional to $\sqrt{g}$, whereas $\boldsymbol{Q}$ has smoothing parameter roughly proportional to g. (Note that these are the same choices that we made in Section 4.4 except that we have switched the roles of $\boldsymbol{Q}$ and $\boldsymbol{Q}'$. So, at a very high level we end up with effectively the same instance of LIP, since $\mathrm{LIP}^{\boldsymbol{Q},\boldsymbol{Q}'}$ is trivially equivalent to $\mathrm{LIP}^{\boldsymbol{Q}',\boldsymbol{Q}}$. However, as we will soon see, we will consider much larger values of g here—namely subexponentially large $g = 2^{n^\delta}$ for some $\delta > 0$—while in Section 4.4 we were able to take $g = \widetilde{\Theta}(n)$.)

From Theorem 6.5, the security of the scheme follows if $\mathcal{D}_s$-$\mathrm{acLIP}^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard and

$$r \geq \eta_\varepsilon(\boldsymbol{Q}') .$$

See [DvW22] for much more discussion of this particular version of LIP with these particular choices of $\boldsymbol{Q}$ and $\boldsymbol{Q}'$. (See also [BGPS23, Section 5] and [Duc24].) Here, we simply observe that, as far as we know, the best algorithms on $\mathcal{D}_s$-$\mathrm{acLIP}^{\boldsymbol{Q},\boldsymbol{Q}'}$ work via a (trivial) reduction to the $\sqrt{g}$-approximate Shortest Vector Problem.[23] The best algorithms for this problem run in time $\mathrm{poly}(n) \cdot 2^{\Theta(n \log(n/\log g)/\log g)}$. So, if we take $g < 2^{o(n \log\log n/\log n)}$, then the best known algorithms require superpolynomial time and this problem is plausibly hard. In fact, as we will see, we already get homomorphic operations for polynomial-depth circuits by taking $g = 2^{n^\delta}$ for a small constant $\delta > 0$, in which case the best known algorithm runs in time $2^{\widetilde{\Theta}(n^{1-\delta})}$.

Finishing with the parameters, recall that we need to take $r \geq \eta_\varepsilon(\boldsymbol{Q}')$ for negligible ε in order to have security. With our choice of $\boldsymbol{Q}'$ above, we see from Lemma 2.12 that it suffices to take any $r > \omega(\sqrt{g \log n})$, but we can simply take $r = \sqrt{ng}$, as this will not meaningfully harm our parameters. Theorem 6.5 then tells us that we obtain a level L homomorphic encryption scheme for L up to

$$\frac{\log\left(\frac{1}{10\sqrt{nd_z}r}\right)}{\log(2m+2)} \geq \frac{\log g}{2\log n + 2\log\log q} - O(1) .$$

So, for example, if we take $g = 2^{n^\delta}$ for some constant $\delta > 0$ (and $q \leq 2^{\mathrm{poly}(n)}$), then we can homomorphically evaluate circuits of depth up to $\widetilde{\Omega}(n^\delta)$, for a scheme for which the best known attacks run in time $2^{\widetilde{\Omega}(n^{1-\delta})}$.

Finally, we discuss the parameters s and q (which matter less than the others). The requirement that $r \geq \sqrt{n \log n} s/q$ (which is needed for the sampling step in the encryption algorithm) implies that we must take $q \geq \sqrt{n \log n} s/r = s \cdot \sqrt{\log n / g}$. We also need $s \geq \sqrt{\log(n)\|\boldsymbol{Q}\|} = \mathrm{poly}(n) g$.

[23] One can get slightly faster algorithms by restricting attention to lattices that have dense sublattices, but the improvements are not significant enough to affect the asymptotics as we present them here. Again, see [DvW22, BGPS23, Duc24] for more discussion.

(In fact, we don't truly need this for our particular choice of $\boldsymbol{Q}$ for which sampling from $\mathcal{D}_s(\boldsymbol{Q})$ is possible even with smaller parameters, but we do not bother to observe this.) So, if we are happy to conjecture that the *average-case* problem $\mathcal{D}_s$-acLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard, then we can simply take $s = \mathrm{poly}(n)g$ and $q = \mathrm{poly}(n)\sqrt{g}$.

Taking q and s to be as small as possible makes for a more efficient scheme, since our scheme performs many operations modulo q and since s controls the size of the entries in $\boldsymbol{P}$ (as in Theorem 2.18). However, to invoke the worst-case to average-case reduction in Theorem 2.18, we should take, e.g., $s = 2^n g$. Then, we obtain security under the assumption that wcLIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ is hard. But, we now must take $q \geq s \cdot \sqrt{\log n / g} = 2^n \sqrt{g \log n}$.

All of the above parameter choices are quite similar to what one obtains for the GSW scheme in [GSW13]. In particular, the best known attacks on LIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ use essentially the same algorithms as the best known attacks on [GSW13], with nearly identical parameters for the same choice of L. However, using sophisticated techniques like those discussed in Section 6.6, it is now known how to build leveled FHE from LWE using parameters that are essentially independent of L.[24] See [DvW22, Section 7] for more discussion of the best known attacks on LIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$.

6.6 A note on bootstrapping

We have described our scheme as a *leveled* FHE scheme. To make it truly fully homomorphic, we would need to show how to correctly evaluate arbitrary circuits rather than circuits with bounded depth L. The only known way to build such true FHE schemes is via Gentry's *bootstrapping* technique [Gen09]. Indeed, Gentry essentially showed a bootstrapping framework, which proceeds in two steps.

First, one designs a leveled FHE scheme that is capable of homomorphically evaluating its own decryption circuit(!). Specifically, one wishes to homomorphically evaluate circuits computing functions of the form $f_{\boldsymbol{C}}(\mathsf{sk}) := \mathsf{Dec}(sk, \boldsymbol{C})$, where notice that the ciphertext $\boldsymbol{C}$ is hard-coded into $f_{\boldsymbol{C}}$, and the input is the secret key. Notice that our scheme has a rather simple decryption circuit. Indeed, our scheme is closely modeled after the scheme in [GSW13] which is known to have this property. (And, our scheme has some similarity with the scheme in [BMM25], where the authors claim that the decryption circuit can be evaluated homomorphically, though they do not show this formally.) Furthermore, recall from Remark 6.4 that one can hope to homomorphically evaluate a decryption circuit even if the scheme cannot handle arbitrary circuits whose depth is nearly as large

[24] Indeed, as we mentioned in the introduction and as was observed in [BMM25] the specific version of the hardness of LIP that we (and [BMM25]) use in our security proof is a stronger assumption than the LWE assumption needed in the state-of-the-art FHE scheme in [BV14]. This is true but a bit subtle. More concretely, the hardness of LIP$^{\boldsymbol{Q},\boldsymbol{Q}'}$ that we need for security is parameterized by g, and the LWE assumption is parameterized by a noise parameter α (which is roughly analogous to our noise parameter r). Recall that we require $g \approx n^{2L}$, and our scheme can be broken if one can solve $\sqrt{g} \approx n^L$-approximate lattice problems. Relatively simple constructions of (leveled) FHE from LWE (such as the original GSW scheme [GSW13]) require $\alpha \lesssim n^{-L}$ for roughly the same reason that we require $g \gtrsim n^{2L}$. But, using sophisticated techniques, [BV14] show how to achieve arbitrarily large polynomial L using only $\alpha \approx n^{-1/2-\varepsilon}$, by effectively reducing the problem of homomorphically evaluating large-depth circuits to the problem of homomorphically evaluating the decryption circuit (as in bootstrapping), and by carefully studying the decryption circuit. Regev showed that the LWE assumption holds provided that it is (quantumly) hard to approximate the length of a shortest non-zero vector to within a factor of $\widetilde{O}(n/\alpha)$ [Reg05]. So, if one compares our scheme to the similarly simple constructions that require $\alpha \lesssim n^{-L}$, we find that the two assumptions are incomparable (or at least not known to be comparable). If one compares to the state of the art, one sees that our assumption is stronger than the relevant version of LWE. We leave it as an open problem to make similar techniques work in our setting in order to achieve security under a weaker assumption. This is closely related to the discussion in Section 6.6.

as the decryption circuit's depth, as was done for [GSW13] in [BV14, AP14, DM15]. So, it seems likely that our scheme or a small modification of it would have this property, possibly even with favorable parameters (e.g., $g \leq \text{poly}(n)$ in Section 6.5).

Second, one includes as part of the public key additional ciphertexts $\boldsymbol{C}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{sk}_i)$, which are encryptions of the bits of the secret key(!). By taking a given ciphertext $\boldsymbol{C}$ and running $\boldsymbol{C}' \leftarrow \mathsf{Eval}(f_{\boldsymbol{C}}, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_\ell)$, one obtains a ciphertext $\boldsymbol{C}'$ with the property that $\mathsf{Dec}(\mathsf{sk}, \boldsymbol{C}') = \mathsf{Dec}(\mathsf{sk}, \boldsymbol{C})$, but where the noise size of $\boldsymbol{C}'$ is bounded. So, one can use bootstrapping to "lower the accumulated noise" from the ciphertext $\boldsymbol{C}$.

However, it is not clear whether publishing the ciphertexts $\boldsymbol{C}_i$ can affect security. Schemes with the property that they remain secure even when the adversary is given access to encryptions of the bits of the secret key are called *circularly secure*. As far as we know, standard lattice-based encryption schemes are all circularly secure, but our proofs of security from standard assumptions fail in this setting. The best we know how to do is to show security under *circular* variants of LWE that effectively incorporate such ciphertexts $\boldsymbol{C}_i$ into the assumption.

We imagine that similar ideas work in our setting. However, the authors of the present work are currently quite reluctant to propose a novel assumption of circular security in this setting, since LIP-based cryptography is a new and rapidly developing area.[25]

References

[AB09] Shweta Agrawal and Xavier Boyen. Identity-based encryption from lattices in the standard model. https://crypto.stanford.edu/~xb/ab09/latticeibe.pdf, 2009. 1

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *Eurocrypt*, 2010. 1

[ABD+21] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber. https://pq-crystals.org/kyber/, 2021. 1

[ADCM12] Michel Abdalla, Angelo De Caro, and Karina Mochetti. Lattice-based hierarchical inner product encryption. In *Latincrypt*, 2012. 1

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from Learning with Errors. In *Asiacrypt*, 2011. 1

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *STOC*, 1996. 1

[ALS16] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *Crypto*, 2016. 1

[ALS21] Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. A $2^{n/2}$-time algorithm for $\sqrt{n}$-SVP and $\sqrt{n}$-Hermite SVP, and an improved time-approximation tradeoff for (H)SVP. In *Eurocrypt*, 2021. 11

[25] [BMM25] claim that their scheme can be generically converted into a true FHE scheme via bootstrapping. But, they do not seem to address the question of circular security.

[AP14] Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In *Crypto*, 2014. 1, 34, 37

[APvW25] Bill Allombert, Alice Pellet-Mary, and Wessel van Woerden. Cryptanalysis of rank-2 Module-LIP: A single real embedding is all it takes. In *Eurocrypt*, 2025. 3

[ARW24] Léo Ackermann, Adeline Roux-Langlois, and Alexandre Wallet. Public-key encryption from the lattice isomorphism problem. In *WCC*, 2024. 3, 16

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993. 10, 11

[BBB+26] Huck Bennett, Drisana Bhatia, Jean-François Biasse, Medha Durisheti, Lucas LaBuff, Vincenzo Pallozzi Lavorante, and Philip Waitkevich. Asymptotic improvements to provable algorithms for the code equivalence problem. *IEEE Transactions on Information Theory*, 72:1093–1108, 2026. 3

[BBCK24] Benjamin Benčina, Alessandro Budroni, Jesús-Javier Chi-Domínguez, and Mukul Kulkarni. Properties of lattice isomorphism as a cryptographic group action. In *Post-Quantum Cryptography*, 2024. 3

[BBD+25] Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Huang Yu-Hsuan, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, and Wessel van Woerden. HAWK: A signature scheme inspired by the lattice isomorphism problem. https://hawk-sign.info/, 2025. 3, 16

[BDK+21] Shi Bai, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium. https://pq-crystals.org/dilithium/, 2021. 1

[BF11] Dan Boneh and David Mandell Freeman. Homomorphic signatures for polynomial functions. In *Eurocrypt*, 2011. 1

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *Eurocrypt*, 2014. 1

[BGPS23] Huck Bennett, Atul Ganju, Pura Peetathawatchai, and Noah Stephens-Davidowitz. Just how hard are rotations of $\mathbb{Z}^n$? Algorithms and cryptography with the simplest lattice. In *Eurocrypt*, 2023. 1, 2, 3, 5, 7, 13, 16, 25, 26, 27, 35

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, 2012. 1, 7

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of Learning with Errors. In *STOC*, 2013. 14

[BM21] Tamar Lichter Blanks and Stephen D. Miller. Generating cryptographically-strong random lattice bases and recognizing rotations of $\mathbb{Z}^n$. In *Post-Quantum Cryptography*, 2021. 16

[BMM25] Pedro Branco, Giulio Malavolta, and Zayd Maradni. Fully-homomorphic encryption from lattice isomorphism. In *TCC*, 2025. 3, 5, 6, 7, 12, 16, 19, 36, 37

[BN24] Henry Bambury and Phong Q. Nguyen. Improved provable reduction of NTRU and hypercubic lattices. In *Post-Quantum Cryptography*, 2024. 3

[BN25] Henry Bambury and Phong Q. Nguyen. Cryptanalysis of an efficient signature based on isotropic quadratic forms. In *Post-Quantum Cryptography*, 2025. 3

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, 2011. 1

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, 2014. 1, 6, 34, 36, 37

[BW25] Huck Bennett and Kaung Myat Htay Win. Relating code equivalence to other isomorphism problems. *Designs, Codes and Cryptography*, 93:701–723, 2025. 3

[CDLP13] Kai-Min Chung, Daniel Dadush, Feng-Hao Liu, and Chris Peikert. On the lattice smoothing parameter problem. In *CCC*, 2013. 11

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *Eurocrypt*, 2010. 1

[CME+25] Clémence Chevignard, Guilhem Mureau, Thomas Espitau, Alice Pellet-Mary, Heorhii Pliatsok, and Alexandre Wallet. A reduction from HAWK to the principal ideal problem in a quaternion algebra. In *Eurocrypt*, 2025. 3

[Dad19] Daniel Dadush. On approximating the covering radius and finding dense lattice subspaces. In *STOC*, 2019. 11

[dCBZI+24] Gustavo de Castro Biage, Gustavo Zambonin, Thaís B. Idalino, Daniel Panario, and Ricardo Custodio. A concrete LIP-based KEM with simple lattices. *IEEE Access*, 12:16408–16420, 2024. 3, 16

[DG23] Léo Ducas and Shane Gibbons. Hull attacks on the lattice isomorphism problem. In *PKC*, 2023. 3

[DM15] Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. In *Eurocrypt*, 2015. 1, 34, 37

[DPPvW23] Léo Ducas, Eamonn W. Postlethwaite, Ludo N. Pulles, and Wessel van Woerden. HAWK: Module LIP makes lattice signatures fast, compact and simple. In *Asiacrypt*, 2023. 3, 16

[Duc24] Léo Ducas. Provable lattice reduction of $\mathbb{Z}^n$ with blocksize $n/2$. *Designs, Codes and Cryptography*, 92:909–916, 2024. 3, 35

[DvW22] Léo Ducas and Wessel van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In *Eurocrypt*, 2022. 1, 2, 3, 4, 5, 6, 7, 11, 12, 16, 19, 20, 24, 25, 35, 36

[EWY23] Thomas Espitau, Alexandre Wallet, and Yang Yu. On Gaussian sampling, smoothing parameter and application to signatures. In *Asiacrypt*, 2023. 11

[FHK+20] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON: Fast-Fourier lattice-based compact signatures over NTRU, 2020. https://falcon-sign.info/. 1

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, 2009. 1, 36

[GKP+13] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *STOC*, 2013. 1

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008. 1, 3, 4, 5, 7, 10, 14, 19, 21, 25

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from Learning with Errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Crypto*, 2013. 1, 3, 5, 6, 7, 30, 31, 32, 34, 36, 37

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, 2013. 1

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, 2015. 1

[Gya26] Manoj Gyawali. Rank-2 module-LIP with special matrices. *Information Processing Letters*, 191:106593, 2026. 3

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, 1998. 1

[HR14] Ishay Haviv and Oded Regev. On the lattice isomorphism problem. In *SODA*, 2014. 16

[JWL+25a] Kaijie Jiang, Anyu Wang, Hengyi Luo, Guoxiao Liu, Gang Tang, Yanbin Pan, and Xiaoyun Wang. Re-randomize and extract: A novel commitment construction framework based on group actions. In *Eurocrypt*, 2025. 3, 16

[JWL+25b] Kaijie Jiang, Anyu Wang, Hengyi Luo, Guoxiao Liu, Yang Yu, and Xiaoyun Wang. Exploiting the symmetry of $\mathbb{Z}^n$: Randomization and the automorphism problem. *Journal of Cryptology*, 38:27, 2025. 3

[KLM+25] Artyom Kuninets, Anton Leevik, Ekaterina Malygina, Evgeniy Melnichuk, and Denis Nabokov. On the construction of Barnes-Wall lattices and their application in cryptography. https://eprint.iacr.org/2025/1640, 2025. 3

[LJPW25] Hengyi Luo, Kaijie Jiang, Yanbin Pan, and Anyu Wang. Cryptanalysis of rank-2 Module-LIP with symplectic automorphisms. In *Asiacrypt*, 2025. 3

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982. 14, 16

[LLM25] Cong Ling, Jingbo Liu, and Andrew Mendelsohn. On the spinor genus and the distinguishing Lattice Isomorphism Problem. In *Asiacrypt*, 2025. 3

[LRvW25] Alberto Leporati, Lorenzo Rovida, and Wessel van Woerden. Beyond LWE: A lattice framework for homomorphic encryption. https://eprint.iacr.org/2025/1171, 2025. 3, 6, 7, 16

[Mic19] Daniele Micciancio. Fully homomorphic encryption from the ground up, 2019. Invited talk at Eurocrypt. 7

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012. 5, 11, 30

[MPPW24] Guilhem Mureau, Alice Pellet-Mary, Georgii Pliatsok, and Alexandre Wallet. Cryptanalysis of rank-2 Module-LIP in totally real number fields. In *Eurocrypt*, 2024. 3

[Mur25] Guilhem Mureau. Special genera of Hermitian lattices and applications to HAWK. In *TCC*, 2025. 3

[Nat24a] National Institute of Standards and Technology. Module-lattice-based digital signature standard. Technical Report Federal Information Processing Standards Publications (FIPS) NIST FIPS 204, U.S. Department of Commerce, Washington, D.C., 2024. 1

[Nat24b] National Institute of Standards and Technology. Module-lattice-based key-encapsulation mechanism standard. Technical Report Federal Information Processing Standards Publications (FIPS) NIST FIPS 203, U.S. Department of Commerce, Washington, D.C., 2024. 1

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case Shortest Vector Problem. In *STOC*, 2009. 1

[Pei16] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10:283–424, 2016. 1

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005. 1, 27, 30, 36

[RS17] Oded Regev and Noah Stephens-Davidowitz. A reverse Minkowski theorem. In *STOC*, 2017. 11

[SLL25] Franciele C. Silva, Maja Lie, and Cong Ling. On hull attacks on the module lattice isomorphism problem. https://eprint.iacr.org/2025/1376, 2025. 3

[Tsa19] Rotem Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. In *Crypto*, 2019. 1

[van25] Wessel van Woerden. Dense and smooth lattices in any genus. In *Asiacrypt*, 2025. 3

[Xag13] Keita Xagawa. Improved (hierarchical) inner-product encryption from lattices. In *PKC*, 2013. 1