Fully-Homomorphic Encryption from Lattice Isomorphism

Pedro Branco
Bocconi University

Giulio Malavolta
Bocconi University

Zayd Maradni
MPI-SWS

Abstract

The lattice isomorphism problem (LIP) asks, given two lattices Λ_0 and Λ_1, to decide whether there exists an orthogonal linear map from Λ_0 to Λ_1. In this work, we show that the hardness of (a circular variant of) LIP implies the existence of a fully-homomorphic encryption scheme for all classical and quantum circuits. Prior to our work, LIP was only known to imply the existence of basic cryptographic primitives, such as public-key encryption or digital signatures.

Contents

1 Introduction

A rank-n lattice Λ is defined as the the set of all integer combinations of n linearly independent vectors $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, which form a basis of the lattice. The study of lattices in computer science started with the celebrated LLL algorithm [LJL82] and was further motivated by Ajtai's connection with cryptography [Ajt96]. At present, lattice-based cryptography is a paradigm of central importance for the design of cryptographic primitives with advanced functionalities, e.g. [GVW13, GKP+13, BGG+14, GKW17, WZ17], cryptanalytic tools [Cop96], and for the foundations of post-quantum [NIS] and quantum [Mah22, Mah23] cryptography.

In particular, lattices enabled the first construction of fully-homomorphic encryption (FHE) [Gen09], which allows arbitrary computation on encrypted data. Subsequent works proposed constructions from different computational assumptions [vGHV10, GH11, BV11a, BV11b, BGV12, Bra12, BV14], with new appealing properties [LTV12, GSW13], and with improved asymptotic efficiency [GHS12, BDGM19, GH19]. With the exception of a single construction based on obfuscation [CLTV15], essentially all known FHE schemes base their security on (variants of) the hardness of solving noisy linear equations, i.e., the learning with errors (LWE) problem. It is known that the LWE problem reduces to the hardness of finding short vectors in (worst-case) lattices [Reg05, Pei09, BLP+13], and therefore the intractability of this problem consistitutes the foundation of security of known FHE schemes.

In this work, we are interested in whether we can use *different* sources of computational intractability to construct FHE schemes. Specifically, we consider the *lattice isomorphism problem* (LIP) [PS97, HR14]: Given two lattices Λ_0 and Λ_1, the goal of LIP is to determine whether the two lattices are isomorphic, i.e., whether there exists an orthogonal linear transformation mapping Λ_0 to Λ_1. LIP has been recently proposed as a new source of computational intractability for basic cryptographic primitives, such as zero-knowledge protocols [HR14, Dv22], public-key encryption [Dv22, BGPS23], and digital signatures [DPPv22]. However, the technical toolkit available for designing LIP-based cryptosystems is still at its infancy, and nowhere close to the one available for LWE-based cryptography.

While the actual hardness of LIP, and in particular its relation with LWE, is still not well-understood, we believe that constructing FHE from LIP is an important goal: On the one hand, (i) it requires us to expand the technical toolkit available for LIP-based cryptosystems, which may find applications in other contexts. On the other hand, (ii) it may serve as a candidate FHE construction even in the (perhaps unlikely) event of a cryptanalytic breakthrough against LWE, broadening the foundations for sources of computational hardness required to build FHE. And finally, from a more applied perspective, (iii) it may lead to the design of more efficient FHE schemes and kickstart new techniques in computation on encrypted data.

1.1 Our Results

We construct a fully-homomorphic encryption scheme, with a reduction to (a circular variant of) LIP [PS97, HR14]. Specifically, we consider the following variant of the problem, formally introduced in [Dv22] and referred to as *distinguish-LIP*:

- Let Λ_0 and Λ_1 be two lattices in the same genus.
- Sample a bit $b \leftarrow\$ \{0,1\}$ and sample a lattice $\widetilde{\Lambda}$ from the equivalence class of lattices isomorphic to Λ_b.

- Given $(\Lambda_0, \Lambda_1, \widetilde{\Lambda})$, no polynomial-time distinguisher can guess b with probability negligibly better than $1/2$.

For technical reasons, we also require one of the two lattices, say Λ_0, to have a basis with a large singular value. For instance, this trivially holds if $\Lambda_0 \equiv g \cdot \mathbb{Z}^n$, which is a viable choice in the context of LIP [BGPS23]. Under this premise, let us state our main result in the following.

Theorem 1 (Informal)**.** *If the distinguish-LIP problem is hard, then:*

- *There exists a linearly homomorphic encryption scheme, that supports an arbitrary number of linear operations.*
- *Under the additional assumption that the above scheme is secure when encrypting its own secret key (circular security), there exists a homomorphic encryption scheme for all functions.*

The additional circularity assumption is standard in the context of FHE, since it is a critical component to enable bootstrapping [Gen09]. To the best of our knowledge, with the exception of obfuscation-based constructions [CLTV15, BDJ+24], all known FHE schemes rely on this assumption one way or another.

We explicitly mention here that the parameters of distinguish-LIP that we use in this work imply the hardness of LWE (we discuss this more in Section 1.2). Therefore, our theorem statement as-is does not imply the existence of FHE from a new hardness assumption. Nevertheless, we view our result as a promising avenue that may potentially lead to schemes from new assumptions. This optimism is justified by the following reasons: (i) The techniques that we develop diverge significantly from previous LWE-based schemes, and (ii) it is entirely plausible that the parameters that we require are an artifact of the security proof and, to the best of our knowledge, an attack against LWE does not obviously imply an attack against our scheme. Overall, we view our work as a promising first step towards FHE from different sources of hardness.

Going beyond classical circuits, we show that the hardness of distinguish-LIP also implies the existence of FHE capable of homomorphic evaluation of quantum circuits (QFHE) [BJ15, DSS16, Mah23, Bra18, CDM21, GV24]. Recent work [GV24, BK25] shows that a QFHE scheme can be constructed from any classical FHE (with decryption circuit in NC^1, which holds for our scheme), plus an object called *oblivious state preparation* (OSP). Loosely speaking, an OSP allows a classical client to delegate the preparation of a computational basis state or a Hadamard basis state to a quantum server, without leaking which state. Thus, the following theorem suffices to establish the existence of a QFHE scheme from distinguish LIP.

Theorem 2 (Informal)**.** *If the distinguish-LIP problem is hard, then there exists a two-message oblivious state preparation protocol.*

As a direct corollary, we obtain a QFHE scheme from distinguish-LIP. Here we highlight that for our OSP alone we can use much tighter parameters, which in particular do not, to the best of our current understanding, imply the hardness of LWE. Given the wide applicability of OSP as a cryptographic object [BK25], we believe that our OSP protocol is of independent interest. Once again, our approach is quite different from claw-free functions based on LWE [BCM+18].

1.2 On the Hardness of Distinguish-LIP

As alluded to earlier, the lattices Λ_0 and Λ_1 that we consider in our instance of distinguish-LIP are such that their first minima differ by a (large) polynomial factor $g(n) \in n^{O(1)}$ in the lattice dimension n, i.e., $\lambda_1(\Lambda_1)/\lambda_1(\Lambda_0) = g(n) \geq n^{O(1)}$. This gap is roughly proportional to the amount of the homomorphic operations that ciphertexts would support (before bootstrapping). Since orthogonal maps do not change the size of the shortest vector, our variant of distinguish-LIP of is not harder than solving GapSVP_γ for $\gamma < g$ over $\Lambda := \{\Lambda_0, \Lambda_1\}$, i.e., on input a lattice Λ and real $d > 0$, determine whether $\lambda_1(\Lambda) \leq d$ or $\lambda_1(\Lambda) > \gamma \cdot d$.

It is known that if LWE with secret dimension n (and other appropriate parameters) is easy, then there exists efficient quantum [Reg05] and classical [Pei09, BLP+13] algorithms for GapSVP_γ with $\gamma(n) \in \tilde{O}(n)$, where the classical algorithms have some dimension-modulus tradeoffs. Therefore, assuming the hardness of distinguish-LIP for lattices considered in our setting necessitates assuming the hardness of LWE. In light of this, a few considerations are in order:

- The above attack against distinguish-LIP based on a GapSVP oracle seems to only invalidate the assumption, but not the FHE construction itself. Indeed, our security proof follows the *lossiness* argument from [Dv22]: In the first step of the proof, we make a hybrid switch from our lattice Λ_0, to another lattice Λ_1, with bad decoding capabilities. Then we argue that our message is information-theoretically hidden, when encrypted under Λ_1. Note that the lattice Λ_1 is never actually used in the scheme, but only in the security proof.

  For instance, one could make the tautological assumption that our scheme is secure and, to the best of our knowledge, this assumption might hold even in the presence of a GapSVP oracle (i.e., it could hold even if LWE is false). We leave proving the security of our scheme from a well-established assumption that is not implied by LWE as a fascinating open problem.

- As discussed in [Dv22], there exists parameter choices of LIP, in particular over remarkably decodable lattices, such that LIP is potentially even harder than SIS and LWE. Although we do not know at present how to construct FHE based on LIP in this parameter regime, this possibility is not ruled out either and we view our work as a promising first step in this direction.

- Finally, we mention that our OSP construction relies on a variant of distinguish-LIP with a much smaller gap $g(n) \in O(\sqrt{n})$. Therefore, to the best of our knowledge, it does not imply the hardness of LWE, given known reductions.

1.3 Technical Outline

Before discussing our scheme, it is useful to present the distinguish-LIP problem in a more algorithmic form, to avoid having to deal with infinite objects. Following [Dv22], it is going to be convenient to work with the *quadratic form* $\mathbf{Q} := \mathbf{B}^T\mathbf{B}$ of a lattice $\Lambda_{\mathbf{B}}$, where $\mathbf{B}$ is its basis. In this language, the distinguish-LIP problem states that, given two quadratic forms $\mathbf{Q}_0$ and $\mathbf{Q}_1$, the following distributions are computationally indistinguishable:

$$\mathbf{U}^T\mathbf{Q}_0\mathbf{U} \approx_c \mathbf{U}^T\mathbf{Q}_1\mathbf{U}$$

where $\mathbf{U} \leftarrow\!\$\ \mathsf{GL}_n(\mathbb{Z})$ is a randomly sampled unimodular matrix.[1] Quadratic forms have many efficiently computable properties, such as the determinant, the greatest common divisor, and the parity, that are invariant under unimodular transformations. Quadratic forms that share the same invariants are said to be in the same *genus* [Dv22]. In this overview we shall largely ignore this aspect, but it is good to keep in mind that distinguish-LIP is only plausibly hard for pairs of quadratic forms in the same genus (else trivial attacks apply).

The Base Encryption Scheme. Our starting point is the recently introduced encryption scheme from [Dv22], with a new twist that will enable some basic homomorphic properties. Let $\mathbf{Q} := \mathbf{B}^T\mathbf{B}$ be a quadratic form, where $\mathbf{B}$ is a matrix with the smallest singular value greater than g. Additionally, let p, q be two powers of 2 with $p \ll q$. We present the algorithms of the base scheme in the following.

- (Key Generation) To compute the public key, compute $\mathbf{P} := \mathbf{U}^T\mathbf{Q}\mathbf{U}$ where $\mathbf{U} \leftarrow\!\$\ \mathsf{GL}_n(\mathbb{Z})$ is a randomly sampled unimodular matrix. Additionally, sample a vector $\mathbf{r} \leftarrow \mathbb{Z}_p^n$. The public and secret key of the scheme are

$$\mathsf{pk} := (\mathbf{r}, \mathbf{P}) \quad \text{and} \quad \mathsf{sk} := \mathbf{U}.$$

- (Encryption) To encrypt a message $m \in \{0,1\}$, sample a lattice point $\mathbf{x} \leftarrow\!\$\ \mathcal{D}_{\mathbf{P},\sigma}$ (in its coefficient representation) from a discrete Gaussian over $\mathbf{P}$, with parameter σ. Then compute:

$$\mathbf{y} := 1/q \cdot \mathbf{x} \pmod{\mathbb{Z}^n}$$

  where $\mathbf{y}$ lives in the discretized torus $\mathbb{T}_q := \{0, 1/q, \ldots, (q-1)/q\}$. Let $\mathbf{z} := 1/q \cdot \cdot\mathbf{x} - \mathbf{y} \in \mathbb{Z}^n$, the ciphertext ct is composed by:

$$\mathbf{c}_0 := \mathbf{y} \in \mathbb{T}_q^n \quad \text{and} \quad c_1 := \mathbf{r}^T\mathbf{z} + m \pmod{p} \in \mathbb{Z}_p.$$

  This is precisely the same algorithm as in [Dv22], with the crucial difference that we use $\mathbf{z}$ as the mask for the message (instead of $\mathbf{x}$) and we use a *linear* randomness extractor (modulo p, which is a power of 2).

- (Decryption) To recover the message, proceed as follows:
  - Rotate $\mathbf{U}\mathbf{y}$ and observe that:

$$\mathbf{U}\mathbf{y} = 1/q \cdot \mathbf{U}\mathbf{x} - \mathbf{U}\mathbf{z}$$

    where the second summand is integral and therefore $\mathbf{B}\mathbf{U}\mathbf{z}$ is a lattice point.
  - Bound the norm of the first summand as:

$$\|1/q \cdot \mathbf{U}\mathbf{x}\| \lesssim \frac{1}{q \cdot g}\|\mathbf{B}\mathbf{U}\mathbf{x}\| \lesssim \text{small}$$

    for an appropriate choice of the parameters of the discrete Gaussian.

[1] The actual way $\mathbf{U}$ is sampled is slightly more complex since, of course, $\mathsf{GL}_n(\mathbb{Z})$ is an infinite set. For the sake of simplicity we will stick with this terminology in this overview.

– Since $\mathbf{Uy}$ is in the decoding radius of the lattice, one can recover $\mathbf{Uz}$, and consequently $\mathbf{z}$, running the decoding algorithm of $\mathbf{Q}$.
– Compute $c_0 - \mathbf{r}^T\mathbf{z}$ modulo p to recover the message.

It will actually be more convenient for us to give a slightly different formulation of the decryption algorithm. As part of the secret key, compute a vector $\mathbf{s}$ such that:

$$\underbrace{\mathbf{s}^T\mathbf{U}}_{=:\mathsf{sk}} = \mathbf{r}^T \pmod p$$

which always exists, since $\det(\mathbf{U}) = \pm 1$, and thus it is invertible modulo p. Then decryption proceeds by computing:

$$\begin{aligned} c_1 - \mathsf{Round}(\mathbf{s}^T\mathbf{U}\mathbf{y}) \pmod p &\approx c_1 - \mathsf{Round}(\mathbf{r}^T\mathbf{z}) \pmod p \\ &= \mathbf{r}^T\mathbf{z} + m - \mathbf{r}^T\mathbf{z} \pmod p \\ &= m \end{aligned}$$

where Round rounds to the nearest integer and, crucially, the product $\mathbf{s}^T\mathbf{U}\mathbf{y}$ is computed over the reals (in particular, $\mathbf{s}^T\mathbf{U} \neq \mathbf{r}^T$ over the reals). Correctness follows from the same argument as above, plus an application of the Cauchy-Schwarz inequality to establish that:

$$\left|1/q \cdot \mathbf{s}^T\mathbf{U}\mathbf{x}\right| \leq \|\mathbf{s}\| \cdot \|1/q \cdot \mathbf{U}\mathbf{x}\| \lesssim p \cdot \sqrt{n} \cdot \text{small}$$

where $p \ll q$ and therefore the norm of the noise vector is only marginally increased.

Our security proof follows the same outline as the one in [Dv22]: In an indistingiushable hybrid, we sample $\mathbf{P}$ from a *lossy* quadratic form that entails a dense sublattice, so that some information about $\mathbf{z}$ is lost, when computing $\mathbf{y}$. An important difference though is that establishing that $\mathbf{z}$ has some residual min-entropy will not suffice for us, since p is a power of 2. Instead, we will directly analyze the distribution of the discrete Gaussian and then appeal to suitable generalizations of the leftover hash lemma [ILL89]. We omit details here and we refer the reader to the technical sections.

Why Known Techniques Fail. At this point, it is helpful to pause and see what we have achieved by introducing a linear randomness extractor. It can be indeed verified that our scheme is now *additively homomorphic* modulo p. Given two ciphertexts $\mathsf{ct} = (\mathbf{c}_0, c_1)$ and $\mathsf{ct}' = (\mathbf{c}'_0, c'_1)$ for messages m, m' we have that:

$$(\mathbf{c}_0 + \mathbf{c}'_0, c_1 + c'_1) = (\mathbf{y} + \mathbf{y}', \mathbf{r}^T(\mathbf{z} + \mathbf{z}') + (m + m')) \tag{1}$$

where addition is over the reals on the first component and modulo p in the second. As for decryption correctness, we have that:

$$\mathbf{s}^T\mathbf{U}(\mathbf{y} + \mathbf{y}') \approx \mathbf{s}^T\mathbf{U}(\mathbf{z} + \mathbf{z}') = \mathbf{r}^T\mathbf{U}(\mathbf{z} + \mathbf{z}') \pmod p$$

where the norm of the "noise" grows linearly with the amount of additions performed. Given this property, there is a well-known template in lattice-based cryptography to build FHE, formalized by Micciancio [Mic19], and underlying the mathematics of most known FHE schemes. In essence, this framework can be broken down into the following main steps:

- Begin from a linearly-homomorphic encryption $\mathsf{Enc_L}$ with *(noisy) linear decryption*, where the ciphertexts, plaintexts and secret keys live on the same space.
- To encrypt a message m, the encryptor encrypts $m \cdot \mathsf{sk}$ where sk is the secret key. That is, an encryption of m is $\mathsf{Enc_L}(m \cdot \mathsf{sk})$.
- To multiply two ciphertexts, express multiplication as a linear function. Concretely, compute:

$$\begin{aligned}\mathsf{Enc_L}(m \cdot \mathsf{sk}) \cdot \mathsf{Enc_L}(m' \cdot \mathsf{sk}) &= \mathsf{Enc_L}(m \cdot \mathsf{sk} \cdot \mathsf{Enc_L}(m' \cdot \mathsf{sk})) \\ &\approx \mathsf{Enc_L}(m \cdot \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc_L}(m' \cdot \mathsf{sk}))) \\ &\approx \mathsf{Enc_L}(m \cdot m' \cdot \mathsf{sk} + \text{small})\end{aligned}$$

  where the second equality follows from the (noisy) linear decryption of the underlying LHE.

However, there are a couple of major limitations that prevent us from applying this recipe in our context. First, (i) multiplying by a *large* constant will blow up the noise, and we will lose correctness. Second, and more crucially, (ii) ciphertexts, plaintexts and secret keys live in different spaces. Concretely, ciphertext live in $\mathbb{T}_q^n \times \mathbb{Z}_p$, plaintexts live in $\mathbb{Z}_p$ and secret keys live in $\mathbb{Z}^n$. Since the scheme is homomorphic over $\mathbb{Z}_p$, it is not clear how this can be used to evaluate the (approximate) decryption algorithm *as a linear function.*

Linear Homomorphism. To overcome the above limitation, we will introduce a *virtual plaintext space* that will have enough capacity to evaluate the noisy linear decryption function. To encrypt an element $m \in \mathbb{Z}_{pq}$, we first compute its binary decomposition:

$$(m_1, \ldots, m_\mu) \in \{0,1\}^\mu \quad \text{such that} \quad \sum_{i=1}^{\mu} 2^{i-1} \cdot m_i = m \pmod{pq}$$

and $\mu = \log(pq)$, which is always integral since p and q are powers of two. We then encrypt each bit separately using our base encryption scheme, and the new ciphertext consists of the concatenation of all base ciphertexts:

$$\mathsf{ct} := (\mathsf{ct}_i \leftarrow\!\$\, \mathsf{Enc}(\mathsf{pk}, m_i))_{i=1}^{\mu}\,.$$

Second, we scale up all ciphertexts by q. Concretely, we redefine:

$$\mathsf{ct}_i := q \cdot \mathsf{Enc}(\mathsf{pk}, m_i) = \begin{cases} q \cdot \mathbf{c}_{i,0} = q \cdot \mathbf{y}_i \in \mathbb{Z}_q \\ q \cdot c_{i,1} = q \cdot (\mathbf{r}^T \mathbf{z}_i + m_i) \mod pq \in \mathbb{Z}_{pq} \end{cases}.$$

Observe that the above encoding means that we are now encoding each bit as $q \cdot m_i \in \{0, q\}$. In order to have correctness, we have to slightly modify the decryption algorithm, which now computes $\mathbf{s}^T \mathbf{U}(q \cdot \mathbf{y}_i) \mod pq$, rounds to the nearest multiple of q (instead of rounding to the nearest integer) and, finally, reconstructs $m \in \mathbb{Z}_{pq}$ from its binary decomposition. We denote the algorithms of the modified scheme with plaintext space $\mathbb{Z}_{pq}$ as $(\mathsf{KeyGen_L}, \mathsf{Enc_L}, \mathsf{Dec_L})$.

We can see that the scheme is additively homomorphic over $\mathbb{Z}_{pq}$. By Eq. (1), adding the ciphertexts component-wise, results in the addition of each component (modulo p). Therefore, we have that:

$$\sum_{i=1}^{\mu} 2^{i-1} \cdot (m_i + m'_i \pmod{p}) \pmod{pq} = \sum_{i=1}^{\mu} 2^{i-1} \cdot (m_i + m'_i) \pmod{pq} = m + m' \pmod{pq}$$

where the first equality holds if $m_i + m'_i < p$, so that the sum modulo p does not wrap around. This introduces a new invariant, the *plaintext norm*, that will grow as homomorphic operations progress, and we will have to control to make sure that decryption works correctly.

On the other hand, homomorphic multiplication by a (large) power 2^k comes for free: To multiply a ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_\mu)$ by 2^k we simply move up all the components by k-positions. I.e., we define:

$$\mathsf{ct}' := \left(\underbrace{\mathsf{ct}_0, \ldots, \mathsf{ct}_0}_{k\text{-many}}, \mathsf{ct}_1, \ldots, \mathsf{ct}_{\mu-k} \right).$$

where ct_0 are fresh encryptions of 0. This works because:

$$2^k \cdot m = 2^k \sum_{i=1}^{\mu} 2^{i-1} m_i = \sum_{i=1}^{\mu} 2^{i-1+k} m_i = \sum_{i=1}^{\mu-k} 2^{i-1+k} m_i \pmod{pq}$$

where the last equality follows from the fact that all components multiplied by $2^{\mu+i-1}$ will get annihilated modulo pq. Given addition and multiplication by a power of 2, multiplication by an arbitrary (large) constant follows by binary decomposition plus additive reconstruction.

Homomorphic Multiplication. We are now ready to describe the homomorphic multiplication algorithm for the scheme $(\mathsf{KeyGen}_\mathsf{L}, \mathsf{Enc}_\mathsf{L}, \mathsf{Dec}_\mathsf{L})$ as defined above. Before that, however, we will introduce two additional modifications:

- The public key has an extra component which corresponds to an encryption of the secret key $\mathsf{Enc}(\mathsf{pk}, \mathbf{s}^T\mathbf{U})$ where $\mathbf{s}^T\mathbf{U} \in \mathbb{Z}_{pq}^n$. Note that this introduces a *circularity assumption*, i.e., we require that the scheme remains secure even when encrypting its own secret key.
- To encrypt a message $m \in \mathbb{Z}_p$, we compute:

  $$\mathsf{ct}_\mathsf{msg} := \mathsf{Enc}_\mathsf{L}(\mathsf{pk}, q \cdot m) \quad \text{and} \quad \mathsf{ct}_\mathsf{sk} := \mathsf{Enc}_\mathsf{L}(\mathsf{pk}, m \cdot \mathbf{s}^T\mathbf{U})$$

  i.e., even the message in the *virtual plaintext space* is scaled up by q.

Addition and multiplication by a large constant work as before. In fact, the component ct_sk can be discarded as soon as a single homomorphic operation is performed.

Next, we claim that it suffices to multiply two ciphertexts where we are guaranteed that one of the two ciphertexts is a *fresh* encryption of $m \in \{0, 1\}$, i.e., no prior homomorphic evaluation has been performed on such ciphertext. It is well-known, see e.g., [BV14], that this corresponds to the class of computation of *branching programs*, that contains the complexity class NC^1. Let ct be the fresh ciphertext and $\tilde{\mathsf{ct}}$ the evaluated ciphertext with:

$$\mathsf{ct} := (\mathsf{ct}_\mathsf{msg}, \mathsf{ct}_\mathsf{sk}) \quad \text{and} \quad \tilde{\mathsf{ct}} := \left(\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_\mu\right), \quad \text{where } \tilde{\mathsf{ct}}_i = (\tilde{\mathbf{c}}_{i,0}, \tilde{c}_{i,1}) \in q \cdot \mathsf{Enc}(\mathsf{pk}, \tilde{m}_i)$$

such that $\sum_{i=1}^{\mu} 2^{i-1} \tilde{m}_i = q \cdot \tilde{m}$. The homomorphic multiplication algorithm proceeds as follows:

- First compute the ciphertext $\mathsf{ct}_{i,\mathsf{p}}$ by homomorphically multiplying ct_msg times the constant $\tilde{c}_{i,1}/q$. After this step, $\mathsf{ct}_{i,\mathsf{p}}$ is an encryption (under Enc_L) of:

  $$q \cdot m \cdot (\mathbf{r}^T \tilde{\mathbf{z}}_i + \tilde{m}_i))$$

  by the correctness of multiplication by large constants.

- Next, compute $\mathsf{ct}_{i,\mathsf{rand}}$ by multiplying $\mathsf{ct}_{\mathsf{sk}}$ by the constant $\tilde{\mathbf{c}}_{i,0}$. The resulting ciphertext $\mathsf{ct}_{i,\mathsf{rand}}$ is an encryption (under $\mathsf{Enc}_{\mathsf{L}}$) of:
$$q \cdot m \cdot \mathbf{s}^T \mathbf{U} \tilde{\mathbf{y}}_i$$
again by the correctness of multiplication by large constants.

- Sum the resulting ciphertexts $\mathsf{ct}_{i,\mathsf{p}}$ and $\mathsf{ct}_{i,\mathsf{rand}}$ to obtain $\mathsf{ct}_{i,\mathsf{sum}}$. By the correctness of homomorphic addition, this is a ciphertext (under $\mathsf{Enc}_{\mathsf{L}}$) encrypting:
$$q \cdot m \cdot (\mathbf{r}^T \tilde{\mathbf{z}}_i + \tilde{m}_i) + q \cdot m \cdot \mathbf{s}^T \mathbf{U} \tilde{\mathbf{y}}_i \pmod{pq} = q \cdot m\tilde{m}_i + e_i \pmod{pq}$$
since the product is computed over the subgroup $\mathbb{Z}_p$, and by defining $e_i := qm \cdot \mathbf{s}^T \mathbf{U}(\tilde{\mathbf{y}}_i + \tilde{\mathbf{z}}_i)$, for which the norm can be bounded. Note that the term e_i is a noise that appears for the first time here, and it belongs to the *virtual plaintext space* ($\mathbb{Z}_{pq}$). We therefore have to keep the bound on its norm $|e_i| < q$ as a new invariant of our scheme.

- Note that the resulting plaintext is not yet in the prescribed form. What we would like is to obtain an encryption of $q \cdot m\tilde{m}$ (under $\mathsf{Enc}_{\mathsf{L}}$) but instead we have encryptions of $q \cdot m\tilde{m}_i + e_i$. Simply multiplying by a constant 2^i and reconstructing $q\tilde{m}$ homomorphically does not work since it will increase the error term e_i too much, and it will introduce an extra q^2 term. To solve this problem, we devise a method that allows us to reconstruct $q \cdot m\tilde{m} + \hat{e}$ *without* blowing up the noise. We split this in two cases:
  - When $2^i \geq q$, we can safely homomorphically multiply the i-th ciphertext by the constant $2^i/q$ and this will yield an encryption of $2^i \cdot m\tilde{m}_i + e'_i$ where $e'_i := 2^i/q e_i$ is still small.
  - The more subtle case is when $2^i < q$. We show that a combinatorial trick (similar to the one we use for multiplication by constants) gives us an encryption of the desired value $2^i \cdot m\tilde{m}_i + e'_i$ (for some error term e'_i).

  Finally, summing all these ciphertexts gives us the desired encryption of $q \cdot m\tilde{m} + \hat{e}$.

In our analysis we show that, setting the parameters carefully, we can keep the noise, plaintext, and virtual noise norm under control, and evaluate branching programs of any desired length. We refer to the technical sections for more details.

Bootstrapping. To bootstrap our scheme into an FHE, for all polynomial-size circuits, we just need to argue that it can homomorphically evaluate its decryption circuit. Indeed, all operations performed during decryption (linear functions over $\mathbb{Z}_{pq}$ and rounding to a power of 2) can be performed in NC^1. Hence, we can generically apply the bootstrapping technique of Gentry [Gen09] and obtain an FHE.

Evaluating Quantum Circuits. We briefly explain how to upgrade our scheme to evaluation of quantum circuits. As mentioned before, we can focus on constructing an OSP protocol, which is parametrized by a mode $\mu \in \{0,1\}$ and consists of a single message from the client to the server. We have that:

- If $\mu = 0$, then the server prepares the state $|b\rangle$, for some bit $b \in \{0,1\}$.
- If $\mu = 1$, then the server prepares the state $|0\rangle + (-1)^b |1\rangle$, for some bit $b \in \{0,1\}$.

Furthermore, we require that (i) the bit b is efficiently computable by the client, given some classical information returned by the server, and furthermore (ii) the two modes are computationally indistinguishable. It is known [GV24, BK25] that the existence of an OSP plus an FHE with a decryption algorithm in NC^1 implies the existence of a QFHE scheme, so in this overview we focus on the former.

Similarly to our FHE scheme, we will work with two quadratic forms: A quadratic form $\mathbf{Q}_0$ with good decoding, and a quadratic form $\mathbf{Q}_1$ with a lossy sublattice. In injective mode ($\mu = 0$), the client samples $\mathbf{P}$ to be isomorphic to $\mathbf{Q}_0$, and sends it to the server. The server prepares the superposition:

$$\sum_{\mathbf{x}\in\mathbb{Z}^n} \mathcal{D}_{\mathbf{P},\sigma}(\mathbf{x}) \left|\mathbf{x}\right\rangle$$

using an algorithm from [Bra18]. Then it applies the isometric mapping:

$$\sum_{\mathbf{x}\in\mathbb{Z}^n} \mathcal{D}_{\mathbf{P},\sigma}(\mathbf{x}) \left|\mathbf{x}\right\rangle \to \sum_{\mathbf{x}\in\mathbb{Z}^n} \mathcal{D}_{\mathbf{P},\sigma}(\mathbf{x}) \left|\mathbf{x}, 1/q \cdot \mathbf{x} \pmod{\mathbb{Z}^n}\right\rangle$$

and measures the second register in the computational basis to obtain some $\mathbf{y} \in \mathbb{T}_q^n$. The residual state corresponds to:

$$\sum_{\mathbf{x}:\mathbf{y}=1/q\cdot\mathbf{x} \pmod{\mathbb{Z}^n}} \mathcal{D}_{\mathbf{P},\sigma}(\mathbf{x}) \left|\mathbf{x}\right\rangle . \tag{2}$$

However, since the lattice has good decoding properties, it actually holds that $\mathbf{y}$ uniquely determines $\mathbf{x}$ and thus the above state is actually a basis state $\left|\mathbf{x}\right\rangle$, for some $\mathbf{x}$. The next step is to CNOT a hash of the vector onto a fresh register to obtain:

$$\left|\mathbf{x}\right\rangle \left|\mathcal{H}(\mathbf{x})\right\rangle_t .$$

Tracing out the first register, we obtain a basis state with $b = \mathcal{H}(\mathbf{x})$, which is efficiently computable given the unimodular matrix $\mathbf{U}$, since one can simply decode $\mathbf{y}$ and recompute the hash $\mathcal{H}$.

In lossy mode ($\mu = 1$), the quadratic form $\mathbf{P}$ is sampled to be isomorphic to $\mathbf{Q}_1$. The state preparation proceeds as above, except that the state in Eq. (2) is a superposition of multiple vectors (with possibly different amplitudes), since $\mathbf{y}$ no longer uniquely determines $\mathbf{x}$. To get a clean Hadamard state however, we need a state that is a superposition of *exactly* two vectors, and furthermore they must have the same amplitude. To achieve this, we develop a new *filtering* procedure, where we project the state on a random subspace, defined by a universal hash function. In more detail, we apply the isometric mapping:

$$\sum_{\mathbf{x}:\mathbf{y}=1/q\cdot\mathbf{x} \pmod{\mathbb{Z}^n}} \mathcal{D}_{\mathbf{P},\sigma}(\mathbf{x}) \left|\mathbf{x}\right\rangle \to \sum_{\mathbf{x}:\mathbf{y}=1/q\cdot\mathbf{x} \pmod{\mathbb{Z}^n}} \mathcal{D}_{\mathbf{P},\sigma}(\mathbf{x}) \left|\mathbf{x}, \mathcal{G}(x)\right\rangle$$

where $\mathcal{G} : \mathbb{Z}^n \to \{1, \ldots, m\}$ is a universal hash function. Then we measure the second register in the computational basis to obtain some $\mathbf{m} \in \{1, \ldots, m\}$. The residual state is:

$$\sum_{\substack{\mathbf{x}:\ \mathbf{y}=1/q\cdot\mathbf{x} \pmod{\mathbb{Z}^n} \\ \mathcal{G}(\mathbf{x})=\mathbf{m}}} \mathcal{D}_{\mathbf{P},\sigma}(\mathbf{x}) \left|\mathbf{x}\right\rangle .$$

With some (noticeable) probability, the subspace that survives the filtering consists of (i) precisely two vectors, and moreover (ii) with the same amplitude. Thus, CNOTing it onto the target register will produce the desired output. Repeating this procedure sufficiently many times (and appealing to a Chernoff bound) ensures that at least one of the runs is successful.

2 Cryptographic Preliminaries

We denote by κ the security parameter. We say a function f is negligible in the security parameter κ if $f(\kappa) = \kappa^{-\omega(1)}$. Let X, Y be two distribution ensembles. By $X \approx_c Y$ we mean that X and Y are computationally indistinguishable, whereas $X \approx_\varepsilon Y$ means that their statistical distance (i.e., their ℓ_1 norm) is bounded by ε. We also write $X \approx_s Y$, meaning that the statistical distance is bounded by some (unspecified) negligible function.

Vectors $\mathbf{x}$ and matrices $\mathbf{B}$ are denoted in bold and by default vectors are column vectors. We denote the matrix norm $\|\mathbf{B}\| : \max_i \|\mathbf{b}_i\|$, where $\|\cdot\|$ is the Euclidean norm. We denote by $\mathbb{T}_q$ the discretized torus $\mathbb{T}_q : (1/q\mathbb{Z})/\mathbb{Z}$ and identify it with the set of representatives $\{0, 1/q, \ldots, (q-1)/q\}$. Similarly, we always identify $\mathbb{Z}_p$ with the set of representatives $\{0, 1, \ldots, p-1\}$. We denote by $\mathsf{GL}_n(\mathbb{Z})$ the general-linear group over $\mathbb{Z}$.

We recall a version of the leftover hash lemma (LHL), due to Regev [Reg05], over domains $\mathbb{Z}_p$ where p is not necessarily prime, that applies if the input domain is binary.

Lemma 1 (Generalized Leftover Hash Lemma [Reg05])**.** *Fix a positive integer p and $n = \mathsf{poly}(\kappa)$ and let $\mathbf{A} \leftarrow\!\$\, \mathbb{Z}_p^{n\times m}$ be chosen uniformly at random such that $m > 2\kappa + n\log(p)$. Then*

$$(\mathbf{A}, \mathbf{A}\mathbf{x}) \approx_s (\mathbf{A}, \mathbf{u})$$

where $\mathbf{x} \leftarrow\!\$\, \{0,1\}^m$ and $\mathbf{u} \leftarrow\!\$\, \mathbb{Z}_p^n$.

2.1 Fully-Homomorphic Encryption

We recall the standard definition of fully-homomorphic encryption (FHE).

Definition 1 (FHE)**.** *A fully-homomorphic encryption (FHE) scheme is defined by the following algorithms:*

- $\mathsf{KeyGen}(1^\kappa)$ *takes as input a security parameter κ and outputs a pair of public and secret keys $(\mathsf{pk}, \mathsf{sk})$.*
- $\mathsf{Enc}(\mathsf{pk}, m \in \{0,1\})$ *takes as input a public key pk and a message $m \in \{0,1\}$. It outputs a ciphertext ct.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *takes as input a secret key sk and a ciphertext ct. It outputs a message m.*
- $\mathsf{Eval}(\mathsf{pk}, \mathcal{C}, (\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}))$ *takes as input a public key pk, a circuit $\mathcal{C}$ and a ciphertexts $(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)})$. It outputs a new ciphertext ct.*

An FHE scheme should fulfill the following properties:

- *(Correctness) For all $\kappa \in \mathbb{N}$, all circuits $\mathcal{C} : \{0,1\}^\ell \to \{0,1\}$ and all messages $(m_1, \ldots, m_\ell)$ we have that:*
$$\mathcal{C}(m_1, \ldots, m_\ell) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, \mathcal{C}, (\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)})))$$
where $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\in \mathsf{KeyGen}(1^\kappa)$ and $\mathsf{ct}_i \in \mathsf{Enc}(\mathsf{pk}, m_i)$.
- *(Semantic Security) For all m_0 and m_1 the following distributions are computationally indistinguishable:*
$$\{\mathsf{Enc}(\mathsf{pk}, m_0) : (\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\, \mathsf{KeyGen}(1^\kappa)\} \approx_c \{\mathsf{Enc}(\mathsf{pk}, m_1) : (\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\, \mathsf{KeyGen}(1^\kappa)\}\,.$$

It is also standard to require that the scheme is *compact*, meaning that the size of the evaluated ciphertext should be independent of the size of the circuit. Finally, if the encryption algorithm remains secure even in the presence of a ciphertext encrypting its own secret key, we say that the scheme is *circular* secure.

3 Lattices and Gaussians

We denote a lattice $\Lambda_{\mathbf{B}}$ generated by a basis $\mathbf{B}$ as the set:

$$\Lambda_{\mathbf{B}} := \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}.$$

The dual lattice Λ^* of a lattice Λ is defined by $\Lambda^* := \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T\mathbf{y} \in \mathbb{Z}, \forall\ \mathbf{y} \in \Lambda\right\}$, and note that $(\Lambda^*)^* = \Lambda$. The orthogonal lattice $\Lambda^\perp$ is defined by $\Lambda^\perp := \{\mathbf{y} \in \mathbb{Z}^n : \mathbf{By} = \mathbf{0}\}$. We define the rank of a lattice as the rank of (any of) its basis. We denote by $\lambda_i(\Lambda_{\mathbf{Q}})$ the i-th successive minima, i.e., the smallest value of r such that there exists a set of vectors $\{\mathbf{x} \in \Lambda_{\mathbf{Q}} : \|\mathbf{x}\| \leq r\}$ that spans a space of dimension at least i. Note that $\lambda_1(\Lambda_{\mathbf{Q}})$ is the norm of the shortest vector, according to this definition.

For a basis $\mathbf{B}$ we denote by $\mathbf{B}^*$ its Gram-Schmidt orthogonalization. The quadratic form $\mathbf{Q}$ of a lattice $\Lambda_{\mathbf{B}}$ is defined as $\mathbf{Q} = \mathbf{B}^T\mathbf{B} \in \mathbb{R}^{n\times n}$. In a slight abuse of notation, we also denote by $\Lambda_{\mathbf{Q}}$ the lattice corresponding to the quadratic form $\mathbf{Q}$. Given a quadratic form $\mathbf{Q}$, one can efficiently compute a basis for it by, e.g., the Cholesky decomposition. Let $\mathbf{Q} = \mathbf{B}^T\mathbf{B} \in \mathbb{R}^{n\times n}$ be a quadratic form, we define the Gaussian function over $\mathbf{Q}$ with parameter s and centered around $\mathbf{0}$ as:

$$\rho_{\mathbf{Q},s}(\mathbf{x}) := e^{\frac{-\pi\|\mathbf{Bx}\|^2}{s^2}}.$$

Note that setting $\mathbf{B}$ to the identity matrix, we recover the usual probability density function of a Gaussian. The *discrete Gaussian* distribution $\mathcal{D}_{\mathbf{Q},s}$ over $\mathbf{Q}$, with parameter s is given by the probability distribution $\rho_{\mathbf{Q},s}(\mathbf{x})/\rho_{\mathbf{Q},s}(\mathbb{Z}^n)$, where $\rho_{\mathbf{Q},s}(\mathbb{Z}^n) := \sum_{\mathbf{x}\in\mathbb{Z}^n}\rho_{\mathbf{Q},s}(\mathbf{x})$. The following establishes that there exists an efficient sampling algorithm for a discrete Gaussian.

Lemma 2 (Sampling [BLP+13])**.** *There exists a polynomial time algorithm that, given a quadratic form* $\mathbf{Q}$ *with* $\mathrm{rank}(\Lambda_{\mathbf{Q}}) = n$ *and any* $s \geq \|\mathbf{B}^*_{\mathbf{Q}}\| \cdot \sqrt{\ln(2n+4)/\pi}$ *and returns a sample from* $\mathcal{D}_{\mathbf{Q},s}$*.*

The following is a bound on the tails of a discrete Gaussian.

Lemma 3 (Tail Bound [Dv22])**.** *For any quadratic form* $\mathbf{Q}$*, any* $\varepsilon > 0$*, any* $s \geq \eta_\varepsilon(\Lambda_{\mathbf{Q}})$ *we have that*

$$\Pr\left[\|\mathbf{B}^*_{\mathbf{Q}}\mathbf{x}\| > s\sqrt{n} : \mathbf{x} \leftarrow\!\$\ \mathcal{D}_{\mathbf{Q},s}\right] \leq \frac{1+\varepsilon}{1-\varepsilon}\cdot 2^{-n}.$$

The Smoothing Parameter. We recall the definition of the smoothing parameter [MR07].

Definition 2 (Smoothing Parameter [MR07])**.** *For* $\varepsilon > 0$ *and a quadratic form* $\mathbf{Q} \in \mathbb{Z}^{n\times n}$*, the smoothing parameter* $\eta_\varepsilon(\Lambda_{\mathbf{Q}})$ *of the lattice* $\Lambda_{\mathbf{Q}}$ *is the least real* $s > 0$ *such that* $\rho_{\mathbf{Q}^{-1},1/s}(\mathbb{Z}^n\setminus\{\mathbf{0}\}) \leq \varepsilon$*.*

The following lemma gives a bound on the smoothing parameter of a lattice.

Lemma 4 (Smoothing Bound [MR07]). *For any quadratic form* $\mathbf{Q}$ *with* $\operatorname{rank}(\Lambda_{\mathbf{Q}}) = n$ *we have that*

$$\eta_{\varepsilon}(\Lambda_{\mathbf{Q}}) \leq \|\mathbf{B}^*_{\mathbf{Q}}\| \cdot \sqrt{\frac{\ln\left(2n\left(1+\frac{1}{\varepsilon}\right)\right)}{\pi}}$$

for any $\varepsilon > 0$.

The following lemma bounds the smoothing parameter of the orthogonal lattice.

Lemma 5 (Orthogonal Smoothing Bound [BD24]). *Let* $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_p^{n\times m}$ *be uniformly sampled with* $m > 2\kappa + n\log(p)$. *Then*

$$\eta_{\varepsilon}(\Lambda^{\perp}_{\mathbf{A}}) \leq (mn+1)\sqrt{\frac{5\ln\left(2m\left(1+\frac{1}{\varepsilon}\right)\right)}{\pi}}$$

with overwhelming probability.

Proof. Let $\mathbf{A} := (\mathbf{A}_1, \mathbf{A}_2)$, by the leftover hash lemma (Lemma 1) it holds that for a uniformly sampled $\mathbf{A}$, its distribution is statistically close to

$$(\mathbf{A}_1, -\mathbf{A}_1\mathbf{R} + \mathbf{G})$$

with $\mathbf{R} \leftarrow\!\$\ \{0,1\}^{n\times m/2}$ and where $\mathbf{G}$ is the gadget matrix [MP12]. It is well-known that matrices with a trapdoor admit a short basis $\mathbf{S}_{\mathbf{A}}$ for $\Lambda^{\perp}_{\mathbf{A}}$, and it is shown in [MP12] that

$$\|\mathbf{S}^*_{\mathbf{A}}\| \leq (s_1(\mathbf{R})+1)\sqrt{5}$$

where s_1 denotes the largest singular value of a matrix. The desired statement follows by observing that the largest singular value of $\mathbf{R}$ is bounded by $m \cdot n$, and appealing to Lemma 4. □

We also recall a useful lemma from [GPV08].

Lemma 6 (Statistical Distance). *Let* Λ, Λ' *be* n*-dimensional lattices, with* $\Lambda' \subseteq \Lambda$. *Then for any* $\varepsilon \in (0, 1/2)$, *any* $\sigma \geq \eta_{\varepsilon}(\Lambda')$, *and any* $\mathbf{c} \in \mathbb{R}^n$, *the distribution of* $(\mathcal{D}_{\Lambda,\sigma,\mathbf{c}} \bmod \Lambda')$ *is within statistical distance at most* 2ε *of uniform over* $(\Lambda \bmod \Lambda')$.

Finally, we state a slightly generalized version of another lemma from [GPV08] and we present a proof for the sake of completeness.

Lemma 7 (Indistinguishability [GPV08]). *Let* $\mathbf{A} \in \mathbb{Z}_p^{n\times m}$ *such that the columns of* $\mathbf{A}$ *generate* $\mathbb{Z}_p^n$, *let* $0 < \varepsilon < 1/2$, *let* $\sigma \geq \eta_{\varepsilon}(\Lambda^{\perp}_{\mathbf{A}})$, *and let* $\mathbf{c} \in \mathbb{R}^m$ *be arbitrary. Then the following distributions*

$$\mathbf{A}\mathbf{e} \pmod p \approx_{2\varepsilon} \mathbf{u} \pmod p$$

are within statistical distance 2ε, *where* $\mathbf{e} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z}^m,\sigma,\mathbf{c}}$ *and* $\mathbf{u} \leftarrow\!\$\ \mathbb{Z}_p^n$.

Proof. By hypothesis, the set of all syndromes $\{\mathbf{A}\mathbf{e} \pmod p : \mathbf{e} \in \mathbb{Z}^m\} = \mathbb{Z}_p^n$. Now by Lemma 6, for $\mathbf{e} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z}^m,\sigma,\mathbf{c}}$, the distribution of $\mathbf{e} \bmod \Lambda^{\perp}$ is within statistical distance 2ε of uniform over the quotient group $(\mathbb{Z}^m/\Lambda^{\perp})$. Here $\Lambda^{\perp}$ is orthogonal lattice in the sense of p-ary lattice. Because this quotient group is isomorphic to the set of syndromes $\mathbb{Z}_p^n$ via the mapping

$$(\mathbf{e} + \Lambda^{\perp}) \mapsto \mathbf{A}\mathbf{e} \pmod p$$

the claim follows. □

The Lattice Isomorphism Problem. Two lattices are isomorphic if the are related by an orthonormal transformation. In this work we interchangeably talk about lattices and their quadratic form, which is justified since the geometry induced by the quadratic form is precisely equivalent to the one of the corresponding lattice. In terms of quadratic forms $\mathbf{P}$ and $\mathbf{Q}$, isomorphism can be equivalently defined as the existence of a *unimodular* transformation $\mathbf{U} \in \mathsf{GL}_n(\mathbb{Z})$ such that $\mathbf{P} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$. We denote by $[\mathbf{Q}]$ is the equivalence class of quadratic forms isomorphic to $\mathbf{Q}$. In [Dv22], isomorphic equivalence is also defined algorithmically, via a Gaussian distribution of $[\mathbf{Q}]$, denoted by $\mathcal{D}_s([\mathbf{Q}])$. The exact definition will not be important for us, while it suffices to recall the following fact.

Lemma 8 (Gaussian Sampler [Dv22])**.** *Let $\mathbf{Q}$ be a quadratic form with $\mathrm{rank}(\Lambda_{\mathbf{Q}}) = n$ and let*

$$s \geq \max\left\{\lambda_n(\Lambda_{\mathbf{Q}}), \|\mathbf{B}^*_{\mathbf{Q}}\|\sqrt{\ln(2n+4)/\pi}\right\}.$$

Then there exists an expected polynomial-time algorithm, that samples $(\mathbf{R}, \mathbf{U})$ where $\mathbf{R} \leftarrow\!\$\ \mathcal{D}_s([\mathbf{Q}])$ and $\mathbf{U} \in \mathsf{GL}_n(\mathbb{Z})$ such that $\mathbf{R} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$.

The following gives a bound on the matrix norm of a basis sampled through the above procedure.

Lemma 9 (Matrix Norm Bound [Dv22])**.** *For any quadratic form $\mathbf{Q}$ with $\mathrm{rank}(\Lambda_{\mathbf{Q}}) = n$ and any $\varepsilon > 0$ we have that*

$$\Pr\left[\|\mathbf{B}^*_{\mathbf{Q}'}\| > s\sqrt{n} : \mathbf{Q}' \leftarrow\!\$\ \mathcal{D}_s([\mathbf{Q}])\right] \leq \frac{1+\varepsilon}{1-\varepsilon} \cdot 100n \cdot 2^{-n}$$

with $s \geq \max\{\eta_\varepsilon(\Lambda_{\mathbf{Q}}), \lambda_n(\Lambda_{\mathbf{Q}})\}$.

We are now ready the define the distinguish lattice isomorphism problem (LIP).

Definition 3 (Distinguish-LIP [HR14, Dv22])**.** *Given two quadratic forms $\mathbf{Q}_0$ and $\mathbf{Q}_1$ and an $s > 0$, the distinguish-LIP problem postulates that the following distributions are computationally indistinguishable:*

$$\left\{\mathbf{Q}' : \mathbf{Q}' \leftarrow\!\$\ \mathcal{D}_s([\mathbf{Q}_0])\right\} \approx_c \left\{\mathbf{Q}' : \mathbf{Q}' \leftarrow\!\$\ \mathcal{D}_s([\mathbf{Q}_1])\right\}.$$

We refer the reader to [Dv22] for a discussion on the hardness of this problem and for worst-case to average-case reductions. We only mention here that the problem is only plausibly hard if the quadratic forms $\mathbf{Q}_0$ and $\mathbf{Q}_1$ have the same efficiently computable invariants, such as the greatest common divisor, and rational and p-adic equivalence (see [Dv22] for more details). Lattices that are equivalent under such invariants are referred to as being in the same *genus*.

4 Fully Homomorphic Encryption

In the following, we construct our fully-homomorphic encryption scheme (FHE) assuming the hardness of the lattice isomorphism problem.

4.1 The Lattice Family

We consider two families of lattices $\Lambda_{\mathbf{Q}}$ and $\Lambda_{\mathbf{L}}$ defined as follows:

$$\Lambda_{\mathbf{Q}} := g \cdot \mathbb{Z}^{n/2} \oplus \tilde{g} \cdot \mathbb{Z}^{n/2} \quad \text{and} \quad \Lambda_{\mathbf{L}} := \mathbb{Z}^{n/2} \oplus g\tilde{g} \cdot \mathbb{Z}^{n/2}$$

where g and $\tilde{g}$ are such that $\gcd(g, \tilde{g}) = 1$ (i.e., they are co-prime) and we assume without loss of generality that $g < \tilde{g}$. These lattices are generated by the basis:

$$\mathbf{B}_{\mathbf{Q}} := \begin{pmatrix} g \cdot \mathbf{I}_{n/2} & \mathbf{0} \\ \mathbf{0} & \tilde{g} \cdot \mathbf{I}_{n/2} \end{pmatrix} \quad \text{and} \quad \mathbf{B}_{\mathbf{L}} := \begin{pmatrix} \mathbf{I}_{n/2} & \mathbf{0} \\ \mathbf{0} & g\tilde{g} \cdot \mathbf{I}_{n/2} \end{pmatrix}$$

respectively, where $\mathbf{I}_{n/2}$ is the identity matrix of size $n/2$. Consequently, the respective quadratic forms are:

$$\mathbf{Q} := \begin{pmatrix} g^2 \cdot \mathbf{I}_{n/2} & \mathbf{0} \\ \mathbf{0} & \tilde{g}^2 \cdot \mathbf{I}_{n/2} \end{pmatrix} \quad \text{and} \quad \mathbf{L} := \begin{pmatrix} \mathbf{I}_{n/2} & \mathbf{0} \\ \mathbf{0} & g^2\tilde{g}^2 \cdot \mathbf{I}_{n/2} \end{pmatrix}.$$

By construction, we have that $\det(\Lambda_{\mathbf{Q}}) = \det(\Lambda_{\mathbf{L}}) = g^{n/2}\tilde{g}^{n/2}$. Since g and $\tilde{g}$ are co-prime, we also have that $\gcd(\Lambda_{\mathbf{Q}}) = \gcd(\Lambda_{\mathbf{L}}) = 1$ and the same for the parity. Rational and p-adic equivalence are proven using the same argument as in [Dv22]. We define the two unimodular transformations:

$$\mathbf{U}_1 := \begin{pmatrix} g^{-1} \cdot \mathbf{I}_{n/2} & \mathbf{0} \\ \mathbf{0} & g \cdot \mathbf{I}_{n/2} \end{pmatrix} \quad \text{and} \quad \mathbf{U}_2 := \begin{pmatrix} \mathbf{0} & \tilde{g} \cdot \mathbf{I}_{n/2} \\ \tilde{g}^{-1} \cdot \mathbf{I}_{n/2} & \mathbf{0} \end{pmatrix}.$$

Since g and $\tilde{g}$ are integers then their inverse belongs to $\mathbb{Q}$ and thus $\mathbf{U}_1, \mathbf{U}_2 \in \mathsf{GL}_n(\mathbb{Q})$ and we have that $\mathbf{L} = \mathbf{U}_1^T \mathbf{Q} \mathbf{U}_1$ and so the two quadratic forms are equivalent over the rationals. In addition, since g and $\tilde{g}$ are coprime, then for any prime p, we must have that either $\gcd(p, g) = 1$ or $\gcd(p, \tilde{g}) = 1$ (or both), so at least one of them is invertible over the p-adic integers (denoted by $\mathbb{P}_p$ to avoid notation overload) so we have at least one of $\mathbf{U}_1 \in \mathsf{GL}_n(\mathbb{P}_p)$ and $\mathbf{L} = \mathbf{U}_1^T \mathbf{Q} \mathbf{U}_1$ or $\mathbf{U}_2 \in \mathsf{GL}_n(\mathbb{P}_p)$ and $\mathbf{L} = \mathbf{U}_2^T \mathbf{Q} \mathbf{U}_2$ over the p-adics. Thus, the two lattices are in the same genus.

We also mention here that the block structure of the quadratic forms, or the fact that they are integral, does not appear to be strictly necessary for our application, and the only property that we use in our scheme is essentially that the lattice generated by $\mathbf{B}_{\mathbf{Q}}$ has large decoding radius. We nevertheless chose to present this lattice family for the sake of concreteness and we leave exploring generalizations as ground for future work.

4.2 The Base Encryption Scheme

We present a scheme that serves as the fundamental building block for the FHE scheme.

Parameters. Our scheme induces a series of parameters that, for convenience, we list here.

- The rank of the lattice $n := n(\kappa)$ and a constant $\varepsilon \in O(1)$ for the smoothing parameter $\eta_\varepsilon(\Lambda_{\mathbf{Q}})$ of $\Lambda_{\mathbf{Q}}$.
- Two standard deviations $s := s(\kappa)$ and $\sigma := \sigma(\kappa)$ parametrizing the Gaussians used to sample an isomorphic lattice and a vector from the lattice, respectively.
- Two moduli $p := p(\kappa)$ and $q := q(\kappa)$. For convenience we assume that both moduli are powers of 2.

Our analysis will induce the following constraints on the parameters:

- (Efficiency) For sampling the public key (quadratic form) efficiently, we set

$$s \geq \max\left\{\lambda_n(\Lambda_{\mathbf{Q}}), \|\mathbf{B}^*_{\mathbf{Q}}\|\sqrt{\ln(2n+4)/\pi}\right\}. \tag{3}$$

  On the other hand, sampling a vector from the lattice requires us to set

$$\sigma \geq s\sqrt{n}\sqrt{\ln(2n+4)/\pi}. \tag{4}$$

- (Correctness) To invoke the Gaussian tail bound, we set

$$\sigma \geq \eta_\varepsilon(\Lambda_{\mathbf{Q}}) \tag{5}$$

  which also implies that the above inequality holds for all lattices isomorphic to $\Lambda_{\mathbf{Q}}$, since the smoothing parameter is invariant under the lattice isomorphism. We will also require that

$$q \cdot g > 2 \cdot \sigma \cdot p \cdot n. \tag{6}$$

- (Security) For security, we require that

$$\sigma/q \geq (n/2+1)\sqrt{\frac{5\ln\left(n\left(1+2^\kappa\right)\right)}{\pi}} \tag{7}$$

  and, to satisfy the precondition of Lemma 5, that

$$n > 4\kappa + 2\log(p). \tag{8}$$

The Construction. The algorithms our base encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ are defined in the following.

- $\mathsf{KeyGen}(1^\kappa)$: Sample $\mathbf{P}$ and $\mathbf{U} \in \mathsf{GL}_n(\mathbb{Z})$ with $\mathbf{P} \hookleftarrow\$\ \mathcal{D}_s([\mathbf{Q}])$, using the algorithm from Lemma 8. Note that $\mathbf{P} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$. Sample $\mathbf{r} \hookleftarrow\$\ \mathbb{Z}_p^n$, then compute the vector $\mathbf{s} \in \mathbb{Z}_p^n$ such that:

$$\mathbf{s}^T\mathbf{U} = \mathbf{r}^T \pmod p \tag{9}$$

  Output $\mathsf{pk} = (\mathbf{r}, \mathbf{P})$ and $\mathsf{sk} = (\mathbf{s}, \mathbf{U})$.

- $\mathsf{Enc}(\mathsf{pk}, m)$: On input a message $m \in \{0,1\}$, sample $\mathbf{x} \hookleftarrow\$\ \mathcal{D}_{\mathbf{P},\sigma}$ and set

$$\mathbf{y} := 1/q \cdot \mathbf{x} \pmod{\mathbb{Z}^n}. \tag{10}$$

  Define $\mathbf{z} := 1/q \cdot \mathbf{x} - \mathbf{y} \in \mathbb{Z}^n$. Set $\mathsf{ct} := (\mathbf{c}_0 \in \mathbb{T}_q^n, c_1 \in \mathbb{Z}_p)$ where

$$\mathbf{c}_0 := \mathbf{y} \text{ and } c_1 := \mathbf{r}^T\mathbf{z} + m \pmod p.$$

- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: Parse $\mathsf{ct} = (\mathbf{c}_0, c_1)$, then compute

$$\mathbf{s}^T\mathbf{U}\mathbf{c}_0 \in \mathbb{R}$$

  where the computation is done over the reals. Then round to the closest integer, reduce the results modulo p and sum them with c_1.

Before analyzing the scheme, let us first establish that all algorithms are well-defined and run in polynomial time. First, note that a vector $\mathbf{s}$ satisfying Eq. (9) always exists since $\det(\mathbf{U}) = \pm 1$, and consequently $\mathbf{U}$ is invertible modulo p.

Next, we claim that one can efficiently sample from the distributions specified in the algorithms. The efficiency of the key generation procedure follows directly by Lemma 8 along with Eq. (3). On the other hand, for the encryption algorithm, by Lemma 9, we have that with overwhelming probability, the matrix norm of the Gram-Schmidt orthogonalization of the basis corresponding to $\mathbf{P}$ is bounded by $s\sqrt{n}$ and therefore we have that

$$\|\mathbf{B}_{\mathbf{P}}^*\| \sqrt{\ln(2n+4)/\pi} \leq s\sqrt{n}\sqrt{\ln(2n+4)/\pi} \leq \sigma$$

where the last inequality follows by Eq. (4). This implies that we can efficiently sample from $\mathcal{D}_{\mathbf{P},\sigma}$, by Lemma 2.

Correctness. We claim that the decryption algorithm returns the correct message with overwhelming probability. Recall that, rearranging the terms of Eq. (10), we have $\mathbf{y} = 1/q \cdot \mathbf{x} - \mathbf{z}$ for some $\mathbf{z} \in \mathbb{Z}^n$. Substituting to the computation done in the decryption algorithm, we have

$$\mathbf{s}^T\mathbf{U}\mathbf{y} = \mathbf{s}^T\mathbf{U}(1/q \cdot \mathbf{x} - \mathbf{z}) = \underbrace{1/q \cdot \mathbf{s}^T\mathbf{U}\mathbf{x}}_{=:e} - \mathbf{s}^T\mathbf{U}\mathbf{z}$$

over the reals, where the second summand $\mathbf{s}^T\mathbf{U}\mathbf{z}$ is integral, since $\mathbf{z} \in \mathbb{Z}^n$. Observing that the matrix $\mathbf{B}_{\mathbf{Q}}$ is diagonal with entries whose absolute value is greater than g, we have

$$\|1/q \cdot \mathbf{U}\mathbf{x}\| \leq \frac{1}{g} \|1/q \cdot \mathbf{B}_{\mathbf{Q}}\mathbf{U}\mathbf{x}\| = \frac{1}{q \cdot g} \|\mathbf{B}_{\mathbf{P}} \cdot \mathbf{x}\| \leq \frac{\sqrt{n} \cdot \sigma}{q \cdot g}$$

where the last inequality holds with overwhelming probability by appealing to Lemma 3, which we can use by Eq. (5). By the Cauchy-Schwarz inequality, we have that

$$|e| = \left|1/q \cdot \mathbf{s}^T\mathbf{U}\mathbf{x}\right| \leq \|\mathbf{s}\| \cdot \|1/q \cdot \mathbf{U}\mathbf{x}\| \leq \frac{\sigma \cdot p \cdot n}{q \cdot g} < \frac{1}{2}$$

where the last inequality follows by Eq. (6). Thus, rounding to the nearest integer will erase the term e, except with negligible probability. Applying Eq. (9), we have that

$$\mathbf{r}^T\mathbf{z} + m - \mathbf{s}^T\mathbf{U}\mathbf{z} \pmod p = \mathbf{r}^T\mathbf{z} + m - \mathbf{r}^T\mathbf{z} \pmod p = m.$$

Security. We are now ready to prove the security of our encryption scheme. We state our main theorem in the following.

Theorem 3 (Semantic Security)**.** *If the distinguish lattice isomorphism problem is hard for* $\mathbf{Q}$ *and* $\mathbf{L}$*, then the scheme as described above is semantically secure.*

Proof. We consider the following series of hybrid experiments.

- Hybrid $\mathcal{H}_0$: This is the original distribution.
- Hybrid $\mathcal{H}_1$: In this hybrid, we sample $\mathbf{P}$ from $\mathbf{P} \leftarrow\!\!\$\ \mathcal{D}_s([\mathbf{L}])$, instead of $\mathbf{P} \leftarrow\!\!\$\ \mathcal{D}_s([\mathbf{Q}])$, using the algorithm from Lemma 8. By the distinguish-LIP problem (Definition 3) the distributions induced by the two hybrids are computationally indistinguishable.

- Hybrid $\mathcal{H}_2$: In this hybrid, we change the way we sample $\mathbf{r}$. We first sample a random $\mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_p^n$ and compute
$$\mathbf{r}^T = \mathbf{s}^T\mathbf{U} \pmod p.$$
Since $\mathbf{U}$ is invertible modulo p, the distribution of $\mathbf{r}$ is identical to the previous hybrid (uniform over $\mathbb{Z}_p^n$). Thus, this modification is only syntactical and the view of the distinguisher is identical from the previous hybrid.

Observe that in the last hybrid the distribution of $1/q \cdot \mathbf{x}$, conditioned on a fixed $\mathbf{y} \in \mathbb{T}_q^n$, is characterized by the following expression:

$$\frac{1}{q} \cdot \mathbf{x} \leftarrow\!\!\$\ \mathbf{y} - \mathcal{D}_{\mathbf{P},\sigma/q,\mathbf{y}} \tag{11}$$

where $\mathcal{D}_{\mathbf{P},\sigma/q,\mathbf{y}}$ is the discrete Gaussian centered around $\mathbf{y}$, defined by the following (non-normalized) probability density function:

$$\rho_{\mathbf{P},\sigma/q,\mathbf{y}}(\mathbf{x}) := e^{\frac{-\pi\|\mathbf{B_P}(\mathbf{x}-\mathbf{y})\|^2}{(\sigma/q)^2}}$$

with $\mathbf{B_P}$ being a basis for $\mathbf{P}$ (see also Theorem 5.2 in [Dv22]). Next, note that, by definition of $\mathbf{z}$, we can rewrite:

$$\mathbf{z} = 1/q \cdot \mathbf{x} - \mathbf{y} \implies \mathbf{z} \leftarrow\!\!\$\ -\mathcal{D}_{\mathbf{P},\sigma/q,\mathbf{y}}$$

appealing to Eq. (11). Then we claim that $\mathcal{D}_{\mathbf{P},\sigma/q,\mathbf{y}} \equiv \mathbf{U}^{-1} \cdot \mathcal{D}_{\mathbf{L},\sigma/q,\mathbf{Uy}}$, which can be verified by observing that:

$$\rho_{\mathbf{P},\sigma/q,\mathbf{y}}(\mathbf{x}) = e^{\frac{-\pi\|\mathbf{B_P}(\mathbf{x}-\mathbf{y})\|^2}{(\sigma/q)^2}} = e^{\frac{-\pi\|\mathbf{B_L}\mathbf{U}(\mathbf{x}-\mathbf{y})\|^2}{(\sigma/q)^2}} = e^{\frac{-\pi\|\mathbf{B_L}(\mathbf{Ux}-\mathbf{Uy})\|^2}{(\sigma/q)^2}} = \rho_{\mathbf{L},\sigma/q,\mathbf{Uy}}(\mathbf{Ux})$$

where $\mathbf{B_L}$ is the basis for $\mathbf{L}$ and $\mathbf{P} = \mathbf{U}^T\mathbf{L}\mathbf{U}$. Thus $\tilde{\mathbf{z}} := \mathbf{Uz}$ can be equivalently thought of as being sampled from $-\mathbf{U} \cdot \mathcal{D}_{\mathbf{P},\sigma/q,\mathbf{y}} \equiv -\mathcal{D}_{\mathbf{L},\sigma/q,\mathbf{Uy}}$. Parsing $\tilde{\mathbf{z}}$ as the vertical concatenation of $\tilde{\mathbf{z}}_0, \tilde{\mathbf{z}}_1 \in \mathbb{Z}^{n/2}$, and recalling that the basis $\mathbf{B_L}$ of $\Lambda_\mathbf{L}$ is diagonal and its the top left corner is precisely the identity matrix $\mathbf{I}_{n/2}$, we can conclude that $\tilde{\mathbf{z}}_0$ is distributed according to $-\mathcal{D}_{\mathbb{Z}^{n/2},\sigma/q,\tilde{\mathbf{y}}_0}$, where $\tilde{\mathbf{y}}_0$ is the top half of $\mathbf{Uy}$. Next, we rewrite:

$$\begin{aligned} c_1 &= \mathbf{r}^T\mathbf{z} + m \pmod p \\ &= \mathbf{s}^T\mathbf{Uz} + m \pmod p \\ &= \mathbf{s}_0^T\tilde{\mathbf{z}}_0 + \mathbf{s}_1^T\tilde{\mathbf{z}}_1 + m \pmod p \end{aligned}$$

where $\mathbf{s}$ is the vertical concatenation of $\mathbf{s}_0, \mathbf{s}_1 \in \mathbb{Z}_p^{n/2}$. The first summand is therefore distributed according to:

$$\mathbf{s}_0^T\tilde{\mathbf{z}}_0 \pmod p = -\mathbf{s}_0^T\mathcal{D}_{\mathbb{Z}^{n/2},\sigma/q,\tilde{\mathbf{y}}_0} \pmod p.$$

We can conclude the proof by showing that:

$$\mathbf{s}_0^T\mathcal{D}_{\mathbb{Z}^{n/2},\sigma/q,\tilde{\mathbf{y}}_0} \approx_s r : r \leftarrow\!\!\$\ \mathbb{Z}_p$$

since this implies that m is information-theoretically hidden. As the first step, we note that $\mathbf{s}_0^T \in \mathbb{Z}_p^{1\times n/2}$ generates $\mathbb{Z}_p$ if any of its coefficients is odd, which happens with probability $1 - 1/2^{n/2}$.

Second, we observe that, by Eq. (7) and Lemma 5 (where Eq. (8) determines that we satisfy the preconditions), the standard deviation σ/q satisfies:

$$\sigma/q \geq (n/2+1)\sqrt{\frac{5\ln\left(n\left(1+2^{\kappa}\right)\right)}{\pi}} \geq \eta_{2^{-\kappa}}(\Lambda^{\perp}_{\mathbf{s}_0^T}).$$

Finally, statistical indistinguishability follows by Lemma 7. □

4.3 Linear Homomorphic Operations

In the following we show how to compute homomorphically linear operations over the ring $\mathbb{Z}_{pq}$. Before that however, we discuss how to modify our base encryption scheme to encrypt elements of $\mathbb{Z}_{pq}$. To encrypt a message $m \in \mathbb{Z}_{pq}$, we first compute its binary decomposition

$$(m_1, \ldots, m_\mu) \in \{0,1\}^\mu \quad \text{such that} \quad \sum_{i=1}^{\mu} 2^{i-1} \cdot m_i = m \pmod{pq} \tag{12}$$

and $\mu = \log(pq)$, which is always integral since p and q are powers of two. Note that the modular reduction in the sum is redundant at this point, but its purpose will become clear later. We then encrypt each bit separately under our base encryption scheme from Section 4.2, and our ciphertext consists of the concatenation of all base ciphertexts

$$\mathsf{ct} := (\mathsf{ct}_i \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pk}, m_i))_{i=1}^{\mu}\,.$$

It will also be notationally convenient to scale up all ciphertexts by q. Adopting the notation of Section 4.2, we redefine:

$$\mathsf{ct}_i := q \cdot \mathsf{Enc}(\mathsf{pk}, m_i) = \begin{cases} q \cdot \mathbf{c}_{i,0} = q \cdot \mathbf{y}_i \in \mathbb{Z}_q^n \\ q \cdot c_{i,1} = q \cdot (\mathbf{r}^T \mathbf{z}_i + m_i) \pmod{pq} \in \mathbb{Z}_{pq} \end{cases} \tag{13}$$

and observe that the above encoding means that we are now encoding each bit as $q \cdot m_i \in \{0, q\}$. To account for this change, we also need to syntactically modify the decryption algorithm, that now proceeds as follows:

- Compute
$$\mathbf{s}^T\mathbf{U}(q \cdot \mathbf{y}_i) \pmod{pq} = q\mathbf{s}^T\mathbf{U}\mathbf{y}_i \pmod{pq}. \tag{14}$$
- Round to the nearest multiple of q.
- Sum the result with $c_{i,1}$, modulo pq.

Observing that the rounding commutes with the modular reduction, this is precisely the same algorithm as the scheme in Section 4.2 except scaled up by q and, consequently, by the same analysis it returns the correct $q \cdot m_i \in \{0, q\}$ with overwhelming probability. The plaintext is then recovered by rounding to the nearest multiple of q and applying Eq. (12). We denote the algorithms of the modified scheme with plaintext space $\mathbb{Z}_{pq}$ as $(\mathsf{KeyGen}_\mathsf{L}, \mathsf{Enc}_\mathsf{L}, \mathsf{Dec}_\mathsf{L})$. For a given ciphertext $(\mathsf{ct}_i)_{i=1}^{\mu} = \mathsf{ct} \in \mathsf{Enc}_\mathsf{L}(\mathsf{pk}, m)$ satisfying Eq. (13), we will keep track of two quantities:

- (Noise Magnitude) We define the noise magnitude function of a ciphertext as

$$\mathsf{NNorm}(\mathsf{ct}) = \max_i \|q\left(\mathbf{U}\mathbf{y}_i + \mathbf{U}\mathbf{z}_i\right)\|.$$

  Using this notation, decryption succeeds on a ciphertext ct if $\mathsf{NNorm}(\mathsf{ct}) < \frac{q}{2p\sqrt{n}}$.

- (Plaintext Magnitude) We define the plaintext magnitude function of a ciphertext as

$$\mathsf{PtNorm}(\mathsf{ct}) = \max_i \frac{|m_i|}{q}.$$

  Observe that the plaintext magnitude of a freshly encrypted ciphertexts is at most 1.

We are now ready to describe the homomorphic operations for our encryption scheme.

Homomorphic Addition. We now define how the Eval algorithm works for addition over $\mathbb{Z}_{pq}$. On input ℓ ciphertexts $\left(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}\right)$, the evaluation algorithm proceeds as follows.

- $\mathsf{Eval}\left(\mathsf{pk}, +, \left(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}\right)\right)$: Parse $\mathsf{ct}^{(j)}$ as

$$\mathsf{ct}^{(j)} = \left(\mathsf{ct}_1^{(j)}, \ldots, \mathsf{ct}_\mu^{(j)}\right) \quad \text{where } \mathsf{ct}_i^{(j)} = \left(\mathbf{c}_{i,0}^{(j)}, c_{i,1}^{(j)}\right).$$

  Then sum all ciphertexts component-wise modulo pq. That is, return

$$\mathsf{ct} := \left(\sum_{j=1}^{\ell} \mathbf{c}_{i,0}^{(j)} \pmod{pq}, \sum_{j=1}^{\ell} c_{i,1}^{(j)} \pmod{pq}\right)_{i=1}^{\mu}.$$

The following lemma establishes the correctness of the addition operation.

Lemma 10 (Correctness of Addition)**.** *Let* $\left(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}\right)$ *be such that that* $\mathsf{ct}^{(j)} \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, m^{(j)})$*, for some* $m^{(j)} \in \mathbb{Z}_{pq}$*. Furthermore, let* $\mathsf{NNorm}\left(\mathsf{ct}^{(j)}\right) < \frac{q}{2p\sqrt{n\ell}}$ *and* $\mathsf{PtNorm}\left(\mathsf{ct}^{(j)}\right) < p/\ell$*. Then with overwhelming probability, it holds that*

$$\mathsf{Dec}_{\mathsf{L}}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}_{\mathsf{L}}\left(\mathsf{sk}, \mathsf{Eval}\left(\mathsf{pk}, +, \left(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}\right)\right)\right) = \sum_{j=1}^{\ell} m^{(j)} \pmod{pq}$$

and furthermore $\mathsf{NNorm}\left(\mathsf{ct}\right) \leq \sum_{j=1}^{\ell} \mathsf{NNorm}(\mathsf{ct}^{(j)})$ *and* $\mathsf{PtNorm}\left(\mathsf{ct}\right) \leq \sum_{j=1}^{\ell} \mathsf{PtNorm}\left(\mathsf{ct}^{(j)}\right)$.

Proof. Evaluating the expression given by Eq. (14), we can rewrite:

$$\begin{aligned}
\mathbf{s}^T\mathbf{U}\mathbf{c}_{i,0} &= \mathbf{s}^T\mathbf{U}\sum_{j=1}^{\ell}\mathbf{c}_{i,0}^{(j)} \\
&= q \cdot \sum_{j=1}^{\ell} \mathbf{s}^T\mathbf{U}\mathbf{y}_i^{(j)} \pmod{pq} \\
&= q \cdot \mathbf{s}^T\left(\sum_{j=1}^{\ell} \mathbf{U}\mathbf{y}_i^{(j)} + \mathbf{U}\mathbf{z}_i^{(j)} - \mathbf{U}\mathbf{z}_i^{(j)}\right) \pmod{pq}.
\end{aligned}$$

To bound the norm of first two summands, we apply the Cauchy-Schwartz inequality, followed by a triangle inequality, to obtain:

$$\begin{aligned}\left|\mathbf{s}^T q \cdot \left(\sum_{j=1}^{\ell} \mathbf{U}\mathbf{y}_i^{(j)} + \mathbf{U}\mathbf{z}_i^{(j)}\right)\right| &\leq \|\mathbf{s}\| \cdot \left\| q \sum_{j=1}^{\ell} \mathbf{U}\mathbf{y}_i^{(j)} + \mathbf{U}\mathbf{z}_i^{(j)} \right\| \\ &\leq \|\mathbf{s}\| \cdot \sum_{j=1}^{\ell} \left\| q \left(\mathbf{U}\mathbf{y}_i^{(j)} + \mathbf{U}\mathbf{z}_i^{(j)}\right) \right\| \\ &\leq \|\mathbf{s}\| \cdot \ell \cdot \max_j \left\{ \mathsf{NNorm}(\mathsf{ct}^{(j)}) \right\} \\ &< p\sqrt{n} \cdot \frac{q}{2p\sqrt{n}} \\ &= \frac{q}{2}\end{aligned}$$

where the last inequality follows by assumption. This implies that rounding to the nearest multiple of q erases this term. Thus, completing the decryption subroutine, we obtain

$$\begin{aligned}\mathsf{Round}\left(\mathbf{s}^T \mathbf{U}\mathbf{c}_{i,0}\right) + c_{i,1} \pmod{pq} &= -q \cdot \sum_{j=1}^{\ell} \mathbf{s}^T \mathbf{U}\mathbf{z}_i^{(j)} + q \cdot \sum_{j=1}^{\ell} \mathbf{s}^T \mathbf{U}\mathbf{z}_i^{(j)} + m_i^{(j)} \pmod{pq} \\ &= q \cdot \sum_{j=1}^{\ell} m_i^{(j)} \pmod{pq} \\ &= q \cdot \tilde{m}_i \pmod{pq}.\end{aligned}$$

Hence, the decryption algorithm can successfully extract $\tilde{m}_i$ from each ciphertext component ct_i. By assumption on the plaintext magnitude, we know that $\sum_{j=1}^{\ell} m_i^{(j)} < p$, and therefore there is no wrap-around modulo pq in the sum, allowing us to establish that

$$\tilde{m}_i = \sum_{j=1}^{\ell} m_i^{(j)} \tag{15}$$

over the integers. Appealing to Eq. (12) and Eq. (15), we can conclude that the ciphertext ct is an encryption of:

$$\begin{aligned}\sum_{i=1}^{\mu} 2^{i-1} \cdot \tilde{m}_i \pmod{pq} &= \sum_{i=1}^{\mu} 2^{i-1} \cdot \sum_{j=1}^{\ell} m_i^{(j)} \pmod{pq} \\ &= \sum_{j=1}^{\ell} \sum_{i=1}^{\mu} 2^{i-1} m_i^{(j)} \pmod{pq} \\ &= \sum_{j=1}^{\ell} m^{(j)} \pmod{pq}\end{aligned}$$

as desired. The bound on the plaintext magnitude and the noise magnitude of the evaluated ciphertext ct follow by linearity and a triangle inequality. □

Homomorphic Multiplication by a Constant. To complete the linear operations, we define the Eval algorithm to multiply a ciphertext ct by a constant $\delta \in \mathbb{Z}_{pq}$. We formally present our algorithm below.

- $\mathsf{Eval}(\mathsf{pk}, \cdot\delta, \mathsf{ct})$: Parse ct as $(\mathsf{ct}_1, \ldots, \mathsf{ct}_\mu)$, and let $\delta = \sum_{i=1}^{\mu} 2^{i-1} \cdot \delta_i$. For all $i = \{1, \ldots, \mu\}$ define $\mathsf{ct}^{(i)}$ as follows:
  - If $\delta_i = 0$, then set $\mathsf{ct}^{(i)} := (\mathsf{ct}_0, \ldots, \mathsf{ct}_0)$ where $\mathsf{ct}_0 \in q \cdot \mathsf{Enc}(\mathsf{pk}, 0)$ is some fixed encryption of 0 of the base scheme, which can be computed with some fixed *public* randomness.[2]
  - If $\delta_i = 1$, then scale up all the components by $i-1$-positions. I.e., we define

$$\mathsf{ct}^{(i)} := \left(\underbrace{\mathsf{ct}_0, \ldots, \mathsf{ct}_0}_{(i-1)\text{-many}}, \mathsf{ct}_1, \ldots, \mathsf{ct}_{\mu-i+1} \right).$$

  Return $\mathsf{Eval}\left(\mathsf{pk}, +, \left(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\mu)}\right)\right)$.

The following lemma establishes the correctness of the multiplication by constant as described above.

Lemma 11 (Correctness of Multiplication by Constant)**.** *Let* $\mathsf{ct} \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, m)$*, for some* $m \in \mathbb{Z}_{pq}$*, with* $\mathsf{NNorm}(\mathsf{ct}) < \frac{q}{2p\sqrt{n}\mu}$ *and* $\mathsf{PtNorm}(\mathsf{ct}) < p/\mu$*. Then, with overwhelming probability, it holds that*

$$\mathsf{Dec}_{\mathsf{L}}(\mathsf{sk}, \tilde{\mathsf{ct}}) = \mathsf{Dec}_{\mathsf{L}}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, \cdot\delta, \mathsf{ct})) = m \cdot \delta \pmod{pq}$$

and furthermore $\mathsf{NNorm}(\tilde{\mathsf{ct}}) \leq \mathsf{NNorm}(\mathsf{ct}) \cdot \mu$ *and* $\mathsf{PtNorm}(\tilde{\mathsf{ct}}) \leq \mathsf{PtNorm}(\mathsf{ct}) \cdot \mu$*.*

Proof. As a first step, we claim that $\mathsf{ct}^{(i)} \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, m \cdot 2^{i-1} \cdot \delta_i)$. For the case where $\delta_i = 0$, it is clear, whereas for the case where $\delta_i = 1$, one can observe that, by Eq. (12), moving up the ciphertext components by $i-1$ positions, has the same effect as multiplying the underlying message by 2^{i-1}. Furthermore, note that these operations are entirely combinatorial and thus they can only decrease the plaintext and noise magnitude of the ciphertext. Then, appealing to Lemma 10, we have that

$$\begin{aligned}
\mathsf{Dec}_{\mathsf{L}}\left(\mathsf{sk}, \mathsf{Eval}\left(\mathsf{pk}, \cdot\delta, \mathsf{ct}\right)\right) &= \mathsf{Dec}_{\mathsf{L}}\left(\mathsf{sk}, \mathsf{Eval}\left(\mathsf{pk}, +, \left(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\mu)}\right)\right)\right) \\
&= \sum_{i=1}^{\mu} m \cdot 2^{i-1} \cdot \delta_i \pmod{pq} \\
&= m \cdot \delta \pmod{pq}
\end{aligned}$$

as desired, as well as the claimed bounds on the magnitude of the plaintext and of the noise. □

[2] There is no secrecy required for these ciphertexts, so the randomness can be chosen arbitrarily, so long as it is in the support of the correct distribution.

4.4 Fully-Homomorphic Operations

We now show how to extend the scheme presented above into an FHE scheme that supports all homomorphic operations. It will be convenient to consider homomorphic operations modulo 2, and to prove our claim it suffices to show how to implement additions and multiplications between ciphertexts. In fact, it suffices to show how to implement a homomorphic multiplication in the setting where we are guaranteed that one of the two ciphertexts is a *fresh* encryption of $m \in \{0,1\}$, i.e., no prior homomorphic evaluation has been performed on such ciphertext. It is well-known, see e.g., [BV14], that this corresponds to the class of computation of *branching programs*, that contains the complexity class NC^1.

Our main scheme is based on the construction $(\mathsf{KeyGen}_{\mathsf{L}}, \mathsf{Enc}_{\mathsf{L}}, \mathsf{Dec}_{\mathsf{L}})$ described in Section 4.3, which is linearly homomorphic over $\mathbb{Z}_{pq}$. In a slight abuse of notation, we denote the component-wise encryption of vectors and matrices by $\mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, \mathbf{v})$ and $\mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, \mathbf{M})$ and we implement linear operations using the algorithms specified in Section 4.3 in the canonical way. We further adopt the following modifications:

- We augment the public key with an encryption of the secret key $\mathsf{Enc}(\mathsf{pk}, \mathbf{s}^T\mathbf{U})$ where $\mathbf{s}^T\mathbf{U} \in \mathbb{Z}_{pq}^n$. Note that this introduces a *circularity assumption*, i.e., we require that the scheme remains secure even when encrypting its own secret key
- To encrypt a message $m \in \mathbb{Z}_p$, we compute:

$$\mathsf{ct}_{\mathsf{msg}} := \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot m) \quad \text{and} \quad \mathsf{ct}_{\mathsf{sk}} := \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, m \cdot \mathbf{s}^T\mathbf{U}).$$

The first ciphertext component can be computed in the obvious manner, whereas the second ciphertext component can be computed given the secret key. This results into a private-key FHE, but it is well-known that any private-key FHE can be transformed into a public-key one [Rot11]. We expect that other methods would work (e.g., re-randomize the ciphertext) but, to keep things simple, in this work we simple assume that ciphertexts are of this form.

In addition to the plaintext and noise magnitude we will also keep track of a new quantity, which we refer to as the *virtual magnitude* of a ciphertext. In short, we allow the ciphertexts to contain noisy encodings of the message.

- (Virtual Magnitude) We define the virtual magnitude function of a ciphertext as

$$\mathsf{VNorm}(\mathsf{ct}_{\mathsf{msg}}) = |e|$$

  where $\mathsf{ct}_{\mathsf{msg}} \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot m + e)$. Note that the virtual magnitude of freshly encrypted ciphertexts is 0.

We are now ready to describe the homomorphic operations. Homomorphic addition can be performed in a straightforward manner, using the addition algorithm described in Section 4.3. Given two ciphertexts $\mathsf{ct}_0 \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot m_0 + e_0)$ and $\mathsf{ct}_1 \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot m_0 + e_1)$, the algorithm returns:

$$\mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot m_0 + e_0 + q \cdot m_1 + e_1) = \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot (m_0 + m_1) + e_0 + e_1).$$

The circular component of the ciphertexts $\mathsf{ct}_{\mathsf{sk}}$ is simply discarded. Observe that the addition is done over the $\mathbb{Z}_p$ subgroup, since the messages are encoded as multiples of q. Since p is a power of 2, we have that:

$$(m_0 + m_1 \pmod p) \pmod 2 = (m_0 \pmod 2) \oplus (m_1 \pmod 2)$$

as desired. The bound on the plaintext and noise norm follows immediately from Lemma 10, whereas the virtual norm of the evaluated ciphertext is at most the sum of the virtual norms of the individual ciphertexts, by a triangle inequality.

Homomorphic Multiplication. Next we describe an algorithm to homomorphically multiply two ciphertexts. As discussed above, we can assume that one of the two ciphertexts is fresh, i.e., no homomorphic operation has been performed prior to this, and furthermore it encrypts a bit $m \in \{0,1\}$.

- $\mathsf{Eval}\left(\mathsf{pk}, *, \left(\mathsf{ct}, \tilde{\mathsf{ct}}\right)\right)$: Parse ct (the fresh ciphertext) and $\tilde{\mathsf{ct}}$ (the evaluated ciphertext) as

$$\mathsf{ct} := (\mathsf{ct}_{\mathsf{msg}}, \mathsf{ct}_{\mathsf{sk}}) \quad \text{and} \quad \tilde{\mathsf{ct}} := \left(\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_\mu\right), \quad \text{where } \tilde{\mathsf{ct}}_i = (\tilde{\mathbf{c}}_{i,0}, \tilde{c}_{i,1}) \in q \cdot \mathsf{Enc}(\mathsf{pk}, \tilde{m}_i)$$

  such that $\sum_{i=1}^{\mu} 2^{i-1}\tilde{m}_i = q \cdot \tilde{m} + \tilde{e}$. For all $i = \{\log(q/p) + 2, \ldots, \mu\}$ proceed as follows:

  - Compute:

$$\mathsf{ct}_{i,\mathsf{p}} := \mathsf{Eval}\left(\mathsf{pk}, \cdot \tilde{c}_{i,1}/q, \mathsf{ct}_{\mathsf{msg}}\right),$$

    where the division by q is always well-defined since the ciphertext element $\tilde{c}_{i,1}$ is a multiple of q.
  - Compute:

$$\mathsf{ct}_{i,\mathsf{rand}} := \mathsf{Eval}\left(\mathsf{pk}, \cdot \tilde{\mathbf{c}}_{i,0}, \mathsf{ct}_{\mathsf{sk}}\right)$$

    then sum the two ciphertexts to obtain $\mathsf{ct}_{i,\mathsf{sum}} := \mathsf{Eval}(\mathsf{pk}, +, (\mathsf{ct}_{i,\mathsf{p}}, \mathsf{ct}_{i,\mathsf{rand}}))$.
  - If $2^{i-1} = q$, then simply set $\mathsf{ct}_{i,\mathsf{scale}} := \mathsf{ct}_{i,\mathsf{sum}}$. Else, if $2^{i-1} > q$ compute

$$\mathsf{ct}_{i,\mathsf{scale}} := \mathsf{Eval}(\mathsf{pk}, \cdot 2^{i-1}/q, \mathsf{ct}_{i,\mathsf{sum}})$$

    where the division is once again well-defined since q is a power of 2. On the other hand, if $2^{i-1} < q$ then we define

$$\mathsf{ct}_{i,\mathsf{scale}} := \left(\mathsf{ct}_{i,\mathsf{sum},\log(q)-i+2}, \ldots, \mathsf{ct}_{i,\mathsf{sum},\mu}, \mathsf{ct}_0, \ldots, \mathsf{ct}_0\right)$$

    where $\mathsf{ct}_0 \in q \cdot \mathsf{Enc}(\mathsf{pk}, 0)$ is some fixed encryption of 0 of the base scheme.

  Return $\mathsf{ct}_{\mathsf{final}} := \mathsf{Eval}\left(\mathsf{pk}, +, (\mathsf{ct}_{1,\mathsf{scale}}, \ldots, \mathsf{ct}_{\mu,\mathsf{scale}})\right)$.

The following lemma establishes the correctness of the homomorphic multiplication. In a slight abuse of notation, for a fresh ciphertext ct we denote:

- $\mathsf{NNorm}(\mathsf{ct}) = \max\{\mathsf{NNorm}(\mathsf{ct}_{\mathsf{msg}}), \mathsf{NNorm}(\mathsf{ct}_{\mathsf{sk}})\}$,
- $\mathsf{VNorm}(\mathsf{ct}) = \max\{\mathsf{VNorm}(\mathsf{ct}_{\mathsf{msg}}), \mathsf{VNorm}(\mathsf{ct}_{\mathsf{sk}})\}$, and
- $\mathsf{PNorm}(\mathsf{ct}) = \max\{\mathsf{PNorm}(\mathsf{ct}_{\mathsf{msg}}), \mathsf{PNorm}(\mathsf{ct}_{\mathsf{sk}})\}$.

Lemma 12 (Correctness of Multiplication)**.** *Let* $\mathsf{ct} \in (\mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot m), \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, m \cdot \mathbf{s}^T\mathbf{U}))$*, for some* $m \in \{0,1\}$*, with* $\mathsf{NNorm}(\mathsf{ct}) < \frac{q}{2p\sqrt{n(n+1)}\mu^2}$*,* $\mathsf{PtNorm}(\mathsf{ct}) = 1$*, and* $\mathsf{VNorm}(\mathsf{ct}) = 0$*. Let* $\tilde{\mathsf{ct}} \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot \tilde{m} + \tilde{e})$*, for some* $\tilde{m} \in \mathbb{Z}_p$*, with* $\mathsf{PtNorm}(\tilde{\mathsf{ct}}) \leq 2\mu^2(n+1)$ *and* $\mathsf{NNorm}(\tilde{\mathsf{ct}})$ *and* $\mathsf{VNorm}(\tilde{\mathsf{ct}})$ *such that:*

$$\mathsf{VNorm}(\tilde{\mathsf{ct}}) + \mathsf{NNorm}(\tilde{\mathsf{ct}}) \cdot \mu p^2\sqrt{n} + 4q\mu^2(n+1)/p < q/2.$$

Let $\mu^2(n+1) < p$. Then, with overwhelming probability, it holds that:

$$\mathsf{Dec_L}(\mathsf{sk}, \mathsf{ct_{final}}) = \mathsf{Dec_L}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, *, (\mathsf{ct}, \tilde{\mathsf{ct}}))) = qm^* \pmod{pq}$$

where $m^* = m \cdot \tilde{m} \pmod 2$. Furthermore:

- $\mathsf{NNorm}(\mathsf{ct_{final}}) \leq \mu^2(n+1) \cdot \mathsf{NNorm}(\mathsf{ct_{msg}})$.
- $\mathsf{PNorm}(\mathsf{ct_{final}}) \leq \mu^2(n+1)$.
- $\mathsf{VNorm}(\mathsf{ct_{final}}) \leq \mathsf{VNorm}(\tilde{\mathsf{ct}}) + \mathsf{NNorm}(\tilde{\mathsf{ct}}) \cdot \mu p^2\sqrt{n} + 4q\mu^2(n+1)/p$.

Proof. To establish the desired bounds on the quantities of interest, we will track these quantities as the computation progresses in the multiplication algorithm. The following analysis holds for all $i \in \{\log(q/p)+2, \ldots, \mu\}$. By Lemma 11, we can rewrite:

$$\begin{aligned}\mathsf{ct}_{i,\mathsf{p}} &= \mathsf{Eval}\left(\mathsf{pk}, \cdot\tilde{c}_{i,1}/q, \mathsf{ct_{msg}}\right) \\ &= \mathsf{Enc_L}(\mathsf{pk}, q \cdot m \cdot \tilde{c}_{i,1}/q) \\ &= \mathsf{Enc_L}\left(\mathsf{pk}, q \cdot m \cdot \left(\mathbf{r}^T\tilde{\mathbf{z}}_i + \tilde{m}_i\right)\right)\end{aligned}$$

and furthermore:

- $\mathsf{NNorm}(\mathsf{ct}_{i,\mathsf{p}}) \leq \mathsf{NNorm}(\mathsf{ct_{msg}}) \cdot \mu$.
- $\mathsf{PNorm}(\mathsf{ct}_{i,\mathsf{p}}) \leq \mu$.
- $\mathsf{VNorm}(\mathsf{ct}_{i,\mathsf{p}}) = 0$.

Similarly, for the $\mathsf{ct}_{i,\mathsf{rand}}$ term, we obtain:

$$\begin{aligned}\mathsf{ct}_{i,\mathsf{rand}} &= \mathsf{Eval}\left(\mathsf{pk}, \cdot\tilde{\mathbf{c}}_{i,0}, \mathsf{ct_{sk}}\right) \\ &= \mathsf{Enc_L}(\mathsf{pk}, m \cdot \mathbf{s}^T\mathbf{U} \cdot \tilde{\mathbf{c}}_{i,0}) \\ &= \mathsf{Enc_L}(\mathsf{pk}, q \cdot m \cdot \mathbf{s}^T\mathbf{U}\tilde{\mathbf{y}}_i)\end{aligned}$$

then, once again by appealing to Lemma 11, we can bound:

- $\mathsf{NNorm}(\mathsf{ct}_{i,\mathsf{rand}}) \leq \mathsf{NNorm}(\mathsf{ct_{sk}}) \cdot n\mu$.
- $\mathsf{PNorm}(\mathsf{ct}_{i,\mathsf{rand}}) \leq n\mu$.
- $\mathsf{VNorm}(\mathsf{ct}_{i,\mathsf{rand}}) = 0$.

On the other hand, the ciphertext $\mathsf{ct}_{i,\mathsf{sum}}$ contains the sum of the above variables, over $\mathbb{Z}_{pq}$, and by Lemma 10, we can bound:

- $\mathsf{NNorm}(\mathsf{ct}_{i,\mathsf{sum}}) \leq \mathsf{NNorm}(\mathsf{ct_{msg}}) \cdot (n+1)\mu$.
- $\mathsf{PNorm}(\mathsf{ct}_{i,\mathsf{sum}}) \leq (n+1)\mu$.

Where we used the fact that $\mathsf{NNorm}(\mathsf{ct}_{\mathsf{msg}}) = \mathsf{NNorm}(\mathsf{ct}_{\mathsf{sk}})$. Expanding the plaintext of $\mathsf{ct}_{i,\mathsf{sum}}$, we have:

$$\begin{aligned}
&q \cdot m \cdot (\mathbf{r}^T \tilde{\mathbf{z}}_i + \tilde{m}_i) + q \cdot m \cdot \mathbf{s}^T \mathbf{U} \tilde{\mathbf{y}}_i \pmod{pq} \\
&= q \cdot m \cdot (\mathbf{r}^T \tilde{\mathbf{z}}_i + \tilde{m}_i) + q \cdot m \cdot \mathbf{s}^T \mathbf{U} (\tilde{\mathbf{y}}_i + \tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_i) \pmod{pq} \\
&= q \cdot m \cdot (\mathbf{r}^T \tilde{\mathbf{z}}_i + \tilde{m}_i) - q \cdot m \cdot \mathbf{s}^T \mathbf{U} \tilde{\mathbf{z}}_i + q \cdot m \cdot \mathbf{s}^T \mathbf{U} (\tilde{\mathbf{y}}_i + \tilde{\mathbf{z}}_i) \pmod{pq} \\
&= q \cdot m \tilde{m}_i + e_i \pmod{pq}
\end{aligned}$$

where the last equality follows by Eq. (9), since the product is computed over the subgroup $\mathbb{Z}_p$, and by defining $e_i := qm \cdot \mathbf{s}^T \mathbf{U} (\tilde{\mathbf{y}}_i + \tilde{\mathbf{z}}_i)$. We bound the norm of the second summand by:

$$\begin{aligned}
|e_i| &= \left| m \cdot \mathbf{s}^T q \left(\mathbf{U} \tilde{\mathbf{y}}_i + \mathbf{U} \tilde{\mathbf{z}}_i \right) \right| \\
&\leq \left| \mathbf{s}^T q \left(\mathbf{U} \tilde{\mathbf{y}}_i + \mathbf{U} \tilde{\mathbf{z}}_i \right) \right| \\
&\leq \|\mathbf{s}\| \cdot \| q \left(\mathbf{U} \tilde{\mathbf{y}}_i + \mathbf{U} \tilde{\mathbf{z}}_i \right) \| \\
&\leq p\sqrt{n} \cdot \mathsf{NNorm}(\tilde{\mathsf{ct}})
\end{aligned}$$

where the first inequality follows by the fact that $m \in \{0, 1\}$, the second inequality follows by Cauchy-Schwarz, and the third one by the definition of noise norm. Thus, we can conclude that $\mathsf{ct}_{i,\mathsf{sum}} \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot m\tilde{m}_i + e_i)$ with virtual norm $\mathsf{VNorm}(\mathsf{ct}_{i,\mathsf{sum}}) \leq p\sqrt{n} \cdot \mathsf{NNorm}(\tilde{\mathsf{ct}})$.

We now turn to analyze the scaled ciphertext $\mathsf{ct}_{i,\mathsf{scale}}$. We claim that:

$$\mathsf{ct}_{i,\mathsf{scale}} \in \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, 2^{i-1} m\tilde{m}_i + e'_i + \delta_i 2q),$$

for some $\delta_i \in \mathbb{Z}_p$, with:

- $\mathsf{NNorm}(\mathsf{ct}_{i,\mathsf{scale}}) \leq \mathsf{NNorm}(\mathsf{ct}_{i,\mathsf{sum}})$.
- $\mathsf{PNorm}(\mathsf{ct}_{i,\mathsf{scale}}) \leq \mathsf{PNorm}(\mathsf{ct}_{i,\mathsf{sum}})$.
- $\mathsf{VNorm}(\mathsf{ct}_{i,\mathsf{scale}}) \leq p \cdot \mathsf{VNorm}(\mathsf{ct}_{i,\mathsf{sum}})$.

We consider three cases.

- $(2^{i-1} = q)$ In this case $\mathsf{ct}_{i,\mathsf{scale}} = \mathsf{ct}_{i,\mathsf{sum}}$ and therefore all the inequalities follow trivially.
- $(2^{i-1} > q)$ Since $2^{i-1}/q$ is a power of 2, the multiplication by constant does not change the noise norm, nor the plaintext norm of the ciphertext. Furthermore, by Lemma 11, we have

  $$\mathsf{ct}_{i,\mathsf{scale}} = \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, 2^{i-1}/q \cdot (q \cdot m\tilde{m}_i + e_i)) = \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, 2^{i-1} \cdot m\tilde{m}_i + \underbrace{2^{i-1}/q \cdot e_i}_{=:e'_i})$$

  and the bound on the virtual norm follows by observing that $2^{i-1}/q \leq p$.
- $(2^{i-1} < q)$ Recall that, as established above, we can rewrite

  $$\mathsf{ct}_{i,\mathsf{sum}} = \mathsf{Enc}_{\mathsf{L}}(\mathsf{pk}, q \cdot m\tilde{m}_i + e_i)$$

expanding, this ciphertext consists of μ components $(\mathsf{ct}_{i,\mathsf{sum},1}, \ldots, \mathsf{ct}_{i,\mathsf{sum},\mu})$ such that:

$$\mathsf{ct}_{i,\mathsf{sum},j} \in q \cdot \mathsf{Enc}(\mathsf{pk}, m'_j) \quad \text{with} \sum_{j=1}^{\mu} 2^{j-1} \cdot m'_j = q \cdot m\tilde{m}_i + e_i \pmod{pq}$$

by Eq. (12). Then $\mathsf{ct}_{i,\mathsf{scale}}$ is constructed by moving these components down by $\log(q) - i + 2$ positions. Clearly, this operation can only decrease the noise and the plaintext norm of the ciphertext, thus what is left to be shown is a bound on the virtual norm of the ciphertext.

As a thought experiment, consider the ciphertext $\mathsf{ct}'_{i,\mathsf{sum}}$ defined as:

$$\mathsf{ct}'_{i,\mathsf{sum}} := \left(\mathsf{ct}_0, \ldots, \mathsf{ct}_0, \mathsf{ct}_{i,\mathsf{sum},\log(q)-i+2} \ldots, \mathsf{ct}_{i,\mathsf{sum},\mu}\right)$$

that is, the low-order components are substituted by encryptions of 0. By construction, this ciphertext is an encryption of:

$$q \cdot m\tilde{m}_i + e_i - \underbrace{\sum_{j=1}^{\log(q)-i+1} 2^{j-1} \cdot m'_j}_{=:\gamma} = \sum_{j=\log(q)-i+2}^{\mu} 2^{j-1} \cdot m'_j \pmod{pq}.$$

Note that this number is a multiple of $2^{\log(q)-i+1}$, and therefore scaling down the non-zero components (as defined in the computation of $\mathsf{ct}_{i,\mathsf{scale}}$), is equivalent to multiplying by $1/2^{\log(q)-i+1} = 2^{i-1-\log(q)}$ and reducing the modular reduction from pq to

$$pq \cdot 2^{i-1-\log(q)} = 2^{\log(p)+i-1} \geq 2^{\log(q)+1} = 2q$$

where the inequality comes from the fact that $i \geq \log(q) - \log(p) + 2$. Since the difference between $\mathsf{ct}_{i,\mathsf{sum}}$ and $\mathsf{ct}'_{i,\mathsf{sum}}$ is actually erased by the scaling operation, it must be the case that:

$$\begin{aligned}\mathsf{ct}_{i,\mathsf{scale}} &\in \mathsf{Enc}_{\mathsf{L}}\left(\mathsf{pk}, (q \cdot m\tilde{m}_i + e_i - \gamma) \cdot 2^{i-1-\log(q)} + 2q\delta_i\right) \\ &= \mathsf{Enc}_{\mathsf{L}}\Big(\mathsf{pk}, 2^{i-1} \cdot m\tilde{m}_i + \underbrace{(e_i - \gamma) \cdot 2^{i-1-\log(q)}}_{=:e'_i} + 2q\delta_i\Big).\end{aligned}$$

We can then bound the norm of γ by:

$$|\gamma| = \left|\sum_{j=1}^{\log(q)-i+1} 2^{j-1} \cdot m'_j\right| \leq \sum_{j=1}^{\log(q)-i+1} 2^{j-1} \cdot \left|m'_j\right| < p \cdot 2^{\log(q)-i+1}$$

by a triangle inequality and using the fact that $|m'_j| \leq \mathsf{PNorm}(\mathsf{ct}_{i,\mathsf{sum}}) \leq (n+1)\mu \leq p$. Finally

$$|e'_i| = \left|(e_i - \gamma) \cdot 2^{i-1-\log(q)}\right| \leq \left|e_i \cdot 2^{i-1-\log(q)}\right| + p \leq |e_i| + p \leq |e_i| \cdot p$$

for large enough e_i and p, since $2^{i-1-\log(q)} < 1$. Thus, the claimed bound on $\mathsf{VNorm}(\mathsf{ct}_{i,\mathsf{scale}})$.

We can now appeal to Lemma 10, to establish that $\mathsf{ct}_{\mathsf{final}}$ is an encryption of:

$$\begin{aligned}
&\sum_{i=\log(q/p)+2}^{\mu} 2^{i-1} \cdot m\tilde{m}_i + e'_i + \delta_i 2q \\
&= \sum_{i=\log(q/p)+2}^{\mu} 2^{i-1} \cdot m\tilde{m}_i + e'_i + \delta_i 2q + \sum_{i=1}^{\log(q/p)+1} 2^{i-1} \cdot m\tilde{m}_i - \sum_{i=1}^{\log(q/p)+1} 2^{i-1} \cdot m\tilde{m}_i \\
&= q \cdot m\tilde{m} + \tilde{e}m + \sum_{i=\log(q/p)+2}^{\mu} \delta_i 2q + e'_i - \underbrace{\sum_{i=1}^{\log(q/p)+1} 2^{i-1} \cdot m\tilde{m}_i}_{e''}
\end{aligned}$$

with $|e''| \leq 2q/p \cdot \mathsf{PNorm}(\tilde{\mathsf{ct}}) \leq 4q\mu^2(n+1)/p$. Ignoring the "noise" terms, the above plaintext satisfies:

$$\frac{q \cdot m\tilde{m} + 2q\left(\sum_{i=\log(q/p)+2}^{\mu} \delta_i\right)}{q} = m\tilde{m} \pmod 2$$

as desired. Furthermore:

- $\mathsf{NNorm}(\mathsf{ct}_{\mathsf{final}}) \leq \mu \cdot \mathsf{NNorm}(\mathsf{ct}_{i,\mathsf{scale}}) \leq \mu^2(n+1) \cdot \mathsf{NNorm}(\mathsf{ct}_{\mathsf{msg}}) < \frac{q}{2p\sqrt{n}}$.
- $\mathsf{PNorm}(\mathsf{ct}_{\mathsf{final}}) \leq \mu \cdot \mathsf{PNorm}(\mathsf{ct}_{i,\mathsf{scale}}) \leq \mu^2(n+1) < p$.
- $\mathsf{VNorm}(\mathsf{ct}_{\mathsf{final}}) \leq \mathsf{VNorm}(\tilde{\mathsf{ct}}) + \mu \cdot \mathsf{VNorm}(\mathsf{ct}_{i,\mathsf{scale}}) + |e''| \leq \mathsf{VNorm}(\tilde{\mathsf{ct}}) + \mu p^2\sqrt{n} \cdot \mathsf{NNorm}(\tilde{\mathsf{ct}}) + 4q\mu^2(n+1)/p < q/2$.

The first and third conditions guarantee that the decryption algorithm correctly recovers $m\tilde{m}$ modulo 2 from $\mathsf{ct}_{\mathsf{final}}$, concluding our proof. □

Parameters. We propose a set of parameters that satisfies the constraints specified by our basic encryption scheme, while at the same time enabling the homomorphic evaluation of branching programs of and desired depth. We stress that we did not attempt to optimize the choice of parameters for concrete efficiency, and most likely there exist tradeoffs that we do not explore in this work. We claim that the following set of parameters allows us to evaluate a branching program of *any* depth d:

- $n := \mathsf{poly}(\kappa)$ to be a fixed polynomial in the security parameter.
- $p := \mathsf{poly}(\kappa, d, n)$ for a sufficiently large polynomial.
- $q := \mathsf{poly}(p)$ for a sufficiently large polynomial.
- $g := O(q)$ for a sufficiently large constant, to satisfy Eq. (6).
- $\sigma := O(q \cdot n) \cdot \mathsf{poly}(\kappa)$ to satisfy Eqs. (4), (5) and (7).
- $s := O(g) \cdot \mathsf{poly}(\kappa)$ to satisfy Eq. (3).

Note that all parameters are polynomials in the security parameter. To show correctness of homomorphic evaluation, we can assume without loss of generality that the evaluation alternates one homomorphic addition with one homomorphic multiplication (since we can always add 0 and multiply by 1). By Lemma 12, we can see that the noise and plaintext norm of the output ciphertext are bounded by some values that are *independent* of the input ciphertext $\tilde{\mathsf{ct}}$, which means that it suffices to check that our parameters satisfy the pre-conditions imposed by Lemmas 10 and 12. Indeed it holds that:

$$\mathsf{PNorm}(\mathsf{ct}_{\mathsf{final}}) \leq \mu^2(n+1) < p$$

since $\mu = \log(pq) \in O(\kappa)$. Furthermore:

$$\mathsf{NNorm}(\mathsf{ct}_{\mathsf{final}}) \leq \mu^2(n+1) \cdot \mathsf{NNorm}(\mathsf{ct}_{\mathsf{msg}}) \leq \mu^2(n+1) \cdot \frac{\sqrt{n}\sigma}{g} < \frac{q}{2p\sqrt{n}(n+1)\mu^2}$$

where the second inequality follows by the correctness analysis of the base scheme (Section 4.2), scaled up by q. On the other hand, the virtual norm of the output ciphertext grows by an additive factor:

$$\begin{aligned}
\mathsf{VNorm}(\mathsf{ct}_{\mathsf{final}}) &\leq \mathsf{VNorm}(\tilde{\mathsf{ct}}) + \mathsf{NNorm}(\tilde{\mathsf{ct}}) \cdot \mu p^2\sqrt{n} + 4q\mu^2(n+1)/p \\
&\leq \mathsf{VNorm}(\tilde{\mathsf{ct}}) + 2\mu^3 p^2 n^2 \sigma/g + 4q\mu^2(n+1)/p \\
&\leq \mathsf{VNorm}(\tilde{\mathsf{ct}}) + O(\mu^3 p^2 n^3 + q\mu^2 n/p)
\end{aligned}$$

and decryption succeeds as long as the virtual norm is smaller than $q/2$. Since we add a summand $O(\mu^3 p^2 n^3 + q\mu^2 n/p)$ at every operation, for sufficiently large p and q we have that

$$O(\mu^3 p^2 n^3 + q\mu^2 n/p) \cdot d < q/2$$

and decryption will succeed for the evaluation of any d-deep branching program.

4.5 Bootstrapping

The construction we have described above allows one to evaluate homomorphically any branching program of unbounded length. By Barrington's theorem [Bar86], this implies that one can homomorphically compute any NC^1 circuit. Gentry's bootstrapping theorem [Gen09] shows how to convert such a scheme into a fully homomorphic one, i.e., for any polynomial-size circuit, assuming it can homomorphically evaluate its own decryption circuit and provided that an encryption of the secret key is given as part of the public key. Thus, all we need to show is that the decryption algorithm of our scheme can be evaluated by an NC^1 circuit.

The decryption consists of (i) three linear operations (over $\mathbb{Z}_{pq}$) alternated with (ii) rounding to a power of 2. For Boolean circuits, rounding to a power of 2 just means to isolate a particular output wire of the circuit and so it is for free in terms of complexity of the size of the circuit. On the other hand, it is well-known that modular linear operations over κ-bits integers (which is an upper bound on the length of $\mathbb{Z}_{pq}$ elements) are computable by circuits of logarithmic depth, see, e.g., [BV11a] for more details. Thus, we can conclude that our scheme can evaluate its own decryption circuit and therefore it can be bootstrapped into a fully-homomorphic one, following [Gen09].

4.6 A Simple Collision-Resistant Hash Function

In this section, we sketch a direct construction of a collision-resistant hash (CRH) function. While it is well-known that any FHE scheme implies the existence of collision-resistant hashing, we present here a more direct, and certainly more efficient, construction. The hashing key consists of n encryptions of 0 of the base scheme (Section 4.2), that is:

$$\mathsf{hk} := \left\{(\mathbf{y}_i, \mathbf{r}^T\mathbf{z}_i) = \mathsf{ct}_i \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pk}, 0)\right\}_i$$

along with the vector $\mathbf{r}$. To hash a vector $\mathbf{x} \in \{0,1\}^n$, we compute the homomorphic addition:

$$\mathsf{H}(\mathsf{hk}, \mathbf{x}) := \sum_{i=1}^{n} x_i \cdot (\mathbf{y}_i, \mathbf{r}^T\mathbf{z}_i)$$

where the sum is performed component-wise. To prove security, we first move into a hybrid where we replace ct_i by an encryption of 1, for a uniformly chosen $i \leftarrow\!\$\ \{1, \ldots, n\}$. This change goes unnoticed from the CPA security of the underlying encryption scheme. Now, when the adversary provides $\mathbf{x}_0, \mathbf{x}_1 \in \{0,1\}^n$ that break collision resistance of the hash function, first observe that these two vectors are different, so they must differ in at least one position. Let j be such a position. With probability $1/n$ we have that $i = j$. Then we can reach a contradiction as:

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{H}(\mathsf{hk}, \mathbf{x}_0)) \neq \mathsf{Dec}(\mathsf{sk}, \mathsf{H}(\mathsf{hk}, \mathbf{x}_1)).$$

5 Quantum Fully-Homomorphic Encryption

We show how to upgrade our construction to quantum fully-homomorphic encryption (QFHE).

5.1 Quantum Preliminaries

We recall a few basic facts about quantum information and we refer the reader to [NC11] for a comprehensive overview. A (pure) quantum state $|\psi\rangle$ is a unit vector in a separable Hilbert space $\mathcal{H}$. Throughout this work, we will only consider finite-dimensional Hilbert spaces and so we will always assume that $\mathcal{H} \cong \mathbb{C}^d$, for some integer $d \geq 1$. Hilber spaces define physical registers, which we typically denote with a subscript. A Projector-Valued Measure (PVM) consists of a set of projectors $\{\Pi_i\}_i$ that sum up to identity, and if Π_i are not required to be projectors, it is called a Positive Operator-Valued Measure (POVM). Any physical measurement can be described by a POVM, and the Born rule establishes that measuring a state $|\psi\rangle$ will yield outcome i with probability $\langle\psi|\,\Pi_i\,|\psi\rangle$.

The class of efficient quantum algorithm is called BQP, and the notion of computational indistinguishability is extended to BQP algorithms in the natural manner. The notion of *trace distance* is the analogue of statistical distance for quantum states. For the purposes of this work, it suffices to recall that the trace distance can be equivalently defined as the supremum success probability in distinguishing two quantum states, over all (possibly unbounded) quantum operations. Furthermore, the trace distance is non-increasing under quantum operations.

In [Bra18], it is shown how to sample a uniform Gaussian superposition, for a given lattice. We recall the exact statement below.

Theorem 4 (Lattice Superposition Generation [Bra18])**.** *Let $\Lambda_{\mathbf{Q}}$ be an n-dimensional lattice, and let $\sigma \geq \|\mathbf{B}^*_{\mathbf{Q}}\| \cdot \sqrt{\ln(2n+4)/\pi}$. There exists an algorithm that, on input a quadratic form $\mathbf{Q}$ and a precision parameter α, runs in time polynomial in $1/\alpha$ and returns a state within $O(\alpha)$ trace distance from:*

$$\frac{1}{\sqrt{\rho_\sigma(\Lambda_{\mathbf{Q}})}} \sum_{\mathbf{v}\in\Lambda_{\mathbf{Q}}} \rho_{\sqrt{2}\sigma}(\mathbf{v}) \, |\mathbf{v}\rangle \, .$$

5.2 Oblivious State Preparation

In the following we define the notion of oblivious state preparation (OSP), introduced in a recent work [BK25]. Loosely speaking, an OSP allows a classical client to remotely instruct a quantum server to prepare either a computational basis state, or a Hadamard basis state. Furthermore, the two modes should be computationally indistinguishable, even to the eyes of the server. We directly define the round-optimal variant of OSP, where the interaction consists of a single round of messages.

Definition 4 (Oblivious State Preparation [BK25])**.** *An oblivious state preparation (OSP) consists of polynomial-time (classical) algorithm $\mathsf{OSPGen}(1^\lambda, \mu)$ that, on input the security parameter 1^λ and a bit $\mu \in \{0,1\}$, returns a public key pk and a trapdoor td. We require the following properties.*

- *(Correctness) There exists a BQP algorithm that, on input the public key pk computes a state:*

$$|\psi\rangle := \mathsf{H}^\mu \, |b\rangle$$

  for some bit $b \in \{0,1\}$, along with a classical string d. Furthermore, there exists a polynomial-time computable function g such that $g(\mathsf{td}, \mu, d) = b$.

- *(Mode Indistinguishability) The two distributions:*

$$\left\{\mathsf{pk} : (\mathsf{pk}, \mathsf{td}) \leftarrow\!\$\ \mathsf{OSPGen}(1^\lambda, 0)\right\} \approx_c \left\{\mathsf{pk} : (\mathsf{pk}, \mathsf{td}) \leftarrow\!\$\ \mathsf{OSPGen}(1^\lambda, 1)\right\}$$

  are computationally indistinguishable.

It is shown in [BK25], building on the work of [GV24], that a two-message OSP is sufficient to upgrade any (classical) FHE into a QFHE. Thus, we can henceforth concentrate on the task of building a two-message OSP. We recall their theorem below.

Theorem 5 (QFHE from OSP [GV24, BK25])**.** *Given any FHE with decryption in NC^1 and any two-round OSP, there exists a QFHE scheme.*

In our work, we will construct an OSP protocol with a slightly weaker correctness guarantee, namely the server will prepare a state within (an arbitrary small) inverse-polynomial trace distance from the ideal state. Thus, all of our results come with the understanding that the QFHE construction has a small (inverse-polynomial) correctness error.

5.3 Oblivious State Preparation from Lattice Isomorphism

We present our protocol for OSP from the lattice isomorphism problem.

The Lattice Family. We consider the lattices $\Lambda_{\mathbf{Q}}$ and $\Lambda_{\mathbf{L}}$ generated by the basis:

$$\mathbf{B}_{\mathbf{Q}} := \begin{pmatrix} g \cdot \mathbf{I}_{n/2} & \mathbf{0} \\ \mathbf{0} & \tilde{g} \cdot \mathbf{I}_{n/2} \end{pmatrix} \quad \text{and} \quad \mathbf{B}_{\mathbf{L}} := \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & g \cdot \mathbf{I}_{n/2-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{g} \cdot \mathbf{I}_{n/2-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & g\tilde{g} \end{pmatrix}$$

where g and $\tilde{g}$ are co-prime, with $g < \tilde{g}$, and $\tilde{g} \in O(g)$. The fact that they are in the same genus, i.e., they have the same efficiently computable invariants, follows along the same lines as in Section 4.1.

Parameters. Before presenting a formal description of our scheme, we list all the parameters, along with the constraints induced by the scheme.

- The rank of the lattice $n := n(\kappa)$ and a constant $\varepsilon \in O(1)$ for the smoothing parameter $\eta_\varepsilon(\Lambda_{\mathbf{Q}})$ of $\Lambda_{\mathbf{Q}}$.
- Two standard deviations $s := s(\kappa)$ and $\sigma := \sigma(\kappa)$ parametrizing the Gaussians used to sample an isomorphic lattice and a vector from the lattice, respectively.
- A scaling factor $q := q(\kappa)$ that we assume to be even.
- A parameter $m := m(\kappa)$ for the hash function sampled in the scheme.
- A parameter $k := k(\kappa)$ that controls the number of parallel repetitions.

Similarly to the scheme in Section 4.2, we are going to set:

$$s \geq \max\left\{\lambda_n(\Lambda_{\mathbf{Q}}), \|\mathbf{B}^*_{\mathbf{Q}}\|\sqrt{\ln(2n+4)/\pi}\right\} \quad \text{and} \quad \sigma \geq s\sqrt{n}\sqrt{\ln(2n+4)/\pi} \tag{16}$$

so that the sampling procedure is efficient. Then we set:

$$\sigma \geq \eta_\varepsilon(\Lambda_{\mathbf{Q}}) \tag{17}$$

and furthermore:

$$\frac{q \cdot g}{100 \cdot \sqrt{n}} \leq \sigma < \frac{q \cdot g}{2 \cdot \sqrt{2n}} \tag{18}$$

in order to make sure that a Gaussian sample will be in the decoding radius of the lattice $\Lambda_{\mathbf{Q}}$ (the constant on the LHS is arbitrary). Finally, we set $g = \sqrt{n}$ and m, q and k to be sufficiently large polynomials in the security parameter.

The Construction. We present our construction of an OSP in the following.

- $\mathsf{OSPGen}(1^\lambda, \mu)$: Using the algorithm from Lemma 8, sample $\mathbf{P}$ and $\mathbf{U} \in \mathsf{GL}_n(\mathbb{Z})$ as

$$\mathbf{P} := \begin{cases} \mathbf{P} \leftarrow\!\$\, \mathcal{D}_s([\mathbf{Q}]) & \text{if } \mu = 0 \\ \mathbf{P} \leftarrow\!\$\, \mathcal{D}_s([\mathbf{L}]) & \text{if } \mu = 1. \end{cases}$$

  Additionally, sample a $O(\sqrt{n})$-wise independent hash function $\mathcal{G} : \mathbb{Z}^n \to \{1, \ldots, m\}$ and a pairwise independent hash function $\mathcal{H} : \mathbb{Z}^n \to \{0, 1\}$. Set $\mathsf{pk} := (\mathbf{P}, \mathcal{G}, \mathcal{H})$ and $\mathsf{td} := \mathbf{U}$.

The following theorem establishes the mode indistinguishability of the algorithm. The proof is a direct reduction to the distinguishing variant of the lattice isomorphism problem, and it is omitted.

Theorem 6 (Mode Indistinguishability)**.** *If the distinguishing lattice isomorphism problem is hard for* $\mathbf{Q}$ *and* $\mathbf{L}$*, then the scheme as described above is mode indistinguishable.*

BQP Algorithm. We describe a BQP procedure that, on input the public key pk, produces the desired state with overwhelming probability. The algorithm initializes a target qubit $|0\rangle_t$ in the zero state, then for all $i \in \{1, \ldots, k\}$ proceeds as follows:

- Run the algorithm from Theorem 4 to initialize the state:

$$\frac{1}{\sqrt{\rho_\sigma(\Lambda_{\mathbf{P}})}} \sum_{\mathbf{v} \in \Lambda_{\mathbf{P}}} \rho_{\sqrt{2}\sigma}(\mathbf{v}) \, |\mathbf{v}\rangle \, .$$

  For notational convenience, we omit the inverse-polynomial error, and we pretend that the above state was prepared perfectly. By the monotonicity of the trace distance, the error cannot be increased by the subsequent operations.

- Apply the unitary mapping:

$$\frac{1}{\sqrt{\rho_\sigma(\Lambda_{\mathbf{P}})}} \sum_{\mathbf{v} \in \Lambda_{\mathbf{P}}} \rho_{\sqrt{2}\sigma}(\mathbf{v}) \left| \mathbf{B}_{\mathbf{P}}^{-1} \mathbf{v} \right\rangle \propto \sum_{\mathbf{x} \in \mathbb{Z}^n} \mathcal{D}_{\mathbf{P}, \sqrt{2}\sigma}(\mathbf{x}) \, |\mathbf{x}\rangle$$

  omitting normalization factors. Note that the above operation is indeed unitary since the lattice is full rank, and furthermore it can be computed efficiently given any basis $\mathbf{B}_{\mathbf{P}}$.

- Apply the isometric mapping:

$$\sum_{\mathbf{x} \in \mathbb{Z}^n} \mathcal{D}_{\mathbf{P}, \sqrt{2}\sigma}(\mathbf{x}) \, |\mathbf{x}\rangle \to \sum_{\mathbf{x} \in \mathbb{Z}^n} \mathcal{D}_{\mathbf{P}, \sqrt{2}\sigma}(\mathbf{x}) \, |\mathbf{x}, 1/q \cdot \mathbf{x} \pmod{\mathbb{Z}^n}\rangle$$

  and measure the second register in the computational basis to obtain some $\mathbf{y}_i \in \mathbb{T}_q^n$. The residual sate corresponds to:

$$\sum_{\mathbf{x} : \mathbf{y}_i = 1/q \cdot \mathbf{x} \pmod{\mathbb{Z}^n}} \mathcal{D}_{\mathbf{P}, \sqrt{2}\sigma}(\mathbf{x}) \, |\mathbf{x}\rangle \tag{19}$$

  omitting normalization factors.

- Apply the isometric mapping:

$$\sum_{\mathbf{x} : \mathbf{y}_i = 1/q \cdot \mathbf{x} \pmod{\mathbb{Z}^n}} \mathcal{D}_{\mathbf{P}, \sqrt{2}\sigma}(\mathbf{x}) \, |\mathbf{x}\rangle \to \sum_{\mathbf{x} : \mathbf{y}_i = 1/q \cdot \mathbf{x} \pmod{\mathbb{Z}^n}} \mathcal{D}_{\mathbf{P}, \sqrt{2}\sigma}(\mathbf{x}) \, |\mathbf{x}, \mathcal{G}(x)\rangle \tag{20}$$

  which is efficiently computable since $\mathcal{G}$ is. Measure the second register in the computational basis to obtain some $\mathbf{m}_i \in \{1, \ldots, m\}$. The residual state is (again not normalized):

$$\sum_{\substack{\mathbf{x} : \ \mathbf{y}_i = 1/q \cdot \mathbf{x} \pmod{\mathbb{Z}^n} \\ \mathcal{G}(\mathbf{x}) = \mathbf{m}_i}} \mathcal{D}_{\mathbf{P}, \sqrt{2}\sigma}(\mathbf{x}) \, |\mathbf{x}\rangle \, . \tag{21}$$

- Apply coherently the function $\mathcal{H}$ onto a separate register, then CNOT the resulting bit onto the target qubit in register t. Trace out the target qubit from the system, and measure the residual state in the Hadamard basis. Denote the output by $\mathbf{d}_i$.

The algorithm returns the residual state in the t-register, along with the classical strings $\{\mathbf{y}_i, \mathbf{m}_i, \mathbf{d}_i\}_i$.

Analysis. First of all, using the same argument as in Section 4.2, one can show that Eq. (16) implies that of the algorithms are well-defined and run in polynomial time (in particular, that the standard deviation is large enough in order to enable efficient sampling). Next, we analyze the state of the target qubit after a successful completion of the above algorithm. The analysis is completed by proving Lemma 13 and Lemma 14.

Lemma 13 (Case $\mu = 0$)**.** *Let $\mu = 0$ and let $|\psi\rangle_t$ be the state in the t-register after the BQP procedure as described above. Then $|\psi\rangle_t$ is within inverse-polynomial trace distance from:*

$$|\psi\rangle_t \approx_{1/\mathsf{poly}(\kappa)} |x\rangle_t$$

for some $x \in \{0,1\}$ and furthermore x is efficiently computable given the trapdoor and $\{\mathbf{y}_i, \mathbf{m}_i, \mathbf{d}_i\}_i$.

Proof. By Lemma 3 and Eq. (17), the probability of sampling a vector from $\mathcal{D}_{\mathbf{P},\sqrt{2}\sigma}$ with norm:

$$\|\mathbf{B_P} \cdot \mathbf{x}\| \leq \sqrt{2n} \cdot \sigma \tag{22}$$

is negligible. Thus, the state in Eq. (19) is negligibly close (in trace distance) from a state that is entirely supported on vectors $\mathbf{x}$ for which Eq. (22) holds. Recall that:

$$\mathbf{U}\mathbf{y}_i = \mathbf{U}(1/q \cdot \mathbf{x} - \mathbf{z}_i) = 1/q \cdot \mathbf{U}\mathbf{x} - \mathbf{U}\mathbf{z}_i$$

for some integral $\mathbf{z}_i \in \mathbb{Z}^n$. Then, recalling that $\mathbf{B_Q}$ is diagonal with entries greater than g, we have that:

$$\|1/q \cdot \mathbf{U}\mathbf{x}\| \leq \frac{1}{g}\|1/q \cdot \mathbf{B_Q}\mathbf{U} \cdot \mathbf{x}\| = \frac{1}{q \cdot g}\|\mathbf{B_P} \cdot \mathbf{x}\| \leq \frac{\sqrt{2n} \cdot \sigma}{q \cdot g} < \frac{1}{2}$$

by Eq. (18) and Eq. (22). This implies that the noise introduced by the term $1/q \cdot \mathbf{U}\mathbf{x}$ is within the decoding radius of the lattice, which in particular means that $\mathbf{y}_i$ uniquely determines $\mathbf{x}$. Thus, measuring $\mathbf{y}_i$ collapses the state to a basis state $|\mathbf{x}\rangle$, which means that the state in Eq. (19) is negligibly close to a basis state. Consequently, the measurement done in Eq. (20) has no effect on the state, whereas the application of the final CNOT on the target register has the same effect as a regular CNOT, classically controlled on $\mathcal{H}(\mathbf{x})$. We can conclude that the target qubit, after k iterations, is a basis state of the form:

$$|\mathcal{H}(\mathbf{x}_1) \oplus \cdots \oplus \mathcal{H}(\mathbf{x}_k)\rangle_t .$$

Furthermore, the value of each $\mathcal{H}(\mathbf{x}_i)$ can be efficiently recomputed by anyone who knows the trapdoor $\mathbf{U}$ and $\mathbf{y}_i$, by simply running the decoding operation and applying $\mathcal{H}$. □

Lemma 14 (Case $\mu = 1$)**.** *Let $\mu = 1$ and let $|\psi\rangle_t$ be the state in the t-register after the BQP procedure as described above. Then $|\psi\rangle_t$ is within inverse-polynomial trace distance from:*

$$|\psi\rangle_t \approx_{1/\mathsf{poly}(\kappa)} \frac{1}{\sqrt{2}}\left(|0\rangle_t + (-1)^z |1\rangle_t\right)$$

for some $z \in \{0,1\}$ and furthermore x is efficiently computable given the trapdoor and $\{\mathbf{y}_i, \mathbf{m}_i, \mathbf{d}_i\}_i$.

Proof. Recall that the basis $\mathbf{B_L}$ is diagonal and all but one entries are greater than g, so with a similar argument as above, we can establish that all but one coordinates of $\mathbf{U}\mathbf{y}_i$ are within the decoding radius of the one-dimensional sublattice, and thus, they uniquely determine the corresponding coordinate of $\mathbf{U}\mathbf{z}_i$ (as defined above). On the other hand, the top-left corner of the basis $\mathbf{B_L}$ equals 1, and thus the Gaussian tail bound (Lemma 3) bounds the magnitude of the first coordinate of $1/q \cdot \mathbf{U}\mathbf{x}$ to $\sqrt{2n}\sigma/q < g/2$. Substituting $g = \sqrt{n}$, we obtain that there are at most $O(\sqrt{n})$ vectors consistent with the output of the measurement being $\mathbf{y}_i$. This means that the state in Eq. (19) is negligibly close to a state being supported on at most $O(\sqrt{n})$-many basis states. We are now interested in the probability that the following events happen simultaneously:

- (Same Amplitude) The first coordinate of $\mathbf{U}\mathbf{y}_i$ is precisely in-between two lattice points, denoted by their coefficient representation as $\mathbf{x}_{i,0}$ and $\mathbf{x}_{i,1}$. Since the basis $\mathbf{B_L}$ is diagonal, we can equivalently consider sampling each coordinate from a one-dimensional Gaussian, and the above requirement translates to the probability of the fractional part of the sample begin exactly $1/2$. For instance, this happens when the one-dimensional sample equals $q/2$ (i.e., $1/2$ when scaled down). The probability that this event happens is:

$$\frac{\rho_{\sqrt{2}\sigma}(q/2)}{\rho_{\sqrt{2}\sigma}(\mathbb{Z})} \approx \frac{e^{-O(\pi)}}{\sqrt{2}\sigma/\det(\mathbb{Z})} = O\left(\frac{1}{q}\right)$$

  since, by Eq. (18), $\sigma = O(q)$, which is at least inverse polynomial. Note that by symmetry of Gaussians, when this event happens, $\mathbf{x}_{i,0}$ and $\mathbf{x}_{i,1}$ are equally likely to be the lattice point corresponding to $\mathbf{y}_i$ and therefore they have the same amplitude in Eq. (19).

- (Same Filter) It holds that $\mathcal{G}(\mathbf{x}_{i,0}) = \mathcal{G}(\mathbf{x}_{i,1})$ but $\mathcal{G}(\mathbf{x}_{i,0}) \neq \mathcal{G}(\mathbf{x}')$ for any other $\mathbf{x}'$ in the list decoding of $\mathbf{y}_i$. Over the random choice of $\mathcal{G}$, this happens with probability at least:

$$\frac{1}{m^2} \cdot \left(1 - \frac{1}{m}\right)^{O(\sqrt{n})} \geq \frac{1}{m^2} \cdot \left(1 - \frac{O(\sqrt{n})}{m}\right)$$

  since there are at most $O(\sqrt{n})$ vectors in the list decoding of $\mathbf{y}_i$, as argued above. For a large enough m, the above probability is at least inverse-polynomial.

- (Filtering Measurement) The measurement in Eq. (20) returns the outcome $\mathbf{m}_i = \mathcal{G}(\mathbf{x}_{i,0}) = \mathcal{G}(\mathbf{x}_{i,1})$. Since the amplitudes of $\mathbf{x}_{i,0}$ and $\mathbf{x}_{i,1}$ are non-negligible, this event must happen with at least inverse-polynomial probability.

- (Different Hash) It holds that $\mathcal{H}(\mathbf{x}_{i,0}) \neq \mathcal{H}(\mathbf{x}_{i,1})$. By the pairwise independence of $\mathcal{H}$, this happens with probability at least $1/2$.

Overall, the probability that *all of the above* events happen simultaneously is at least inverse-polynomial. By a Chernoff bound, for a large enough k, the probability that there exists an index $i \in \{1, \ldots, k\}$ where all of the above events happen is negligibly close to 1. As argued above, the amplitudes corresponding to $\mathbf{x}_{i,0}$ and $\mathbf{x}_{i,1}$ are identical in Eq. (19). Furthermore, the projection implemented by measuring the last register of the state in Eq. (20) filters out all basis states but $\mathbf{x}_{i,0}$ and $\mathbf{x}_{i,1}$. We can conclude that the state in Eq. (21) is exactly:

$$\frac{1}{\sqrt{2}}\left(|\mathbf{x}_{i,0}\rangle + |\mathbf{x}_{i,1}\rangle\right).$$

We claim that it suffices to consider the action of the above state in the target qubit. To see why, let us analyze the action of applying a CNOT and measuring the register in the Hadamard basis on a generic qubit. For simplicity let us assume that $\mathcal{H}(\mathbf{x}_{i,0}) = 0$ and $\mathcal{H}(\mathbf{x}_{i,1}) = 1$, and the other case follows symmetrically. Applying the CNOT on the target qubit in state $\alpha |0\rangle_t + \beta |1\rangle_t$, we obtain:

$$\frac{1}{\sqrt{2}} \left(|\mathbf{x}_{i,0}\rangle \otimes (\alpha |0\rangle_t + \beta |1\rangle_t) + |\mathbf{x}_{i,1}\rangle \otimes (\alpha |1\rangle_t + \beta |0\rangle_t) \right).$$

Then, applying Hadamard to the first register we are left with:

$$\frac{1}{\sqrt{2}} \left(\sum_{\mathbf{d}} (-1)^{\mathbf{d}\cdot\mathbf{x}_{i,0}} |\mathbf{d}\rangle \otimes (\alpha |0\rangle_t + \beta |1\rangle_t) + \sum_{\tilde{\mathbf{d}}} (-1)^{\tilde{\mathbf{d}}\cdot\mathbf{x}_{i,1}} |\tilde{\mathbf{d}}\rangle \otimes (\alpha |1\rangle_t + \beta |0\rangle_t) \right).$$

Measuring the first register we obtain some $\mathbf{d}_i$ and the residual state is:

$$\begin{aligned}
&\frac{1}{\sqrt{2}} \left((-1)^{\mathbf{d}_i\cdot\mathbf{x}_{i,0}} (\alpha |0\rangle_t + \beta |1\rangle_t) + (-1)^{\mathbf{d}_i\cdot\mathbf{x}_{i,1}} (\alpha |1\rangle_t + \beta |0\rangle_t) \right) \\
&= \frac{1}{\sqrt{2}} \left(((-1)^{\mathbf{d}_i\cdot\mathbf{x}_{i,0}}\alpha + (-1)^{\mathbf{d}_i\cdot\mathbf{x}_{i,1}}\beta) |0\rangle_t + ((-1)^{\mathbf{d}_i\cdot\mathbf{x}_{i,1}}\alpha + (-1)^{\mathbf{d}_i\cdot\mathbf{x}_{i,0}}\beta) |1\rangle_t \right) \\
&= \frac{1}{\sqrt{2}} \left(|0\rangle_t + (-1)^z |1\rangle_t \right)
\end{aligned}$$

where $z := \mathbf{d}_i \cdot \mathbf{x}_{i,0} \oplus \mathbf{d}_i \cdot \mathbf{x}_{i,1} = \mathbf{d}_i \cdot (\mathbf{x}_{i,0} \oplus \mathbf{x}_{i,1})$. Note that z can be efficiently recomputed given the trapdoor $\mathbf{U}$ and $(\mathbf{d}_i, \mathbf{y}_i, \mathbf{m}_i)$, by:

- Finding all of the $O(\sqrt{n})$-many plausible pre-images of $\mathbf{y}_i$. This is efficiently computable, since all coordinates of the vector are uniquely determined, except for the first one. For that coordinate, one simply returns the $O(\sqrt{n})$ integral values closer to the sample.
- Verifying if all of the above events indeed happened for $\mathbf{y}_i$. Once $\mathbf{y}_i$, and $\mathbf{m}_i$ are fixed, all of the conditions are deterministic and efficiently checkable.
- Computing $z := \mathbf{d}_i(\mathbf{x}_{i,0} \oplus \mathbf{x}_{i,1})$.

What is left to be shown is that subsequent applications of the above procedure (CNOT and Hadamard basis measurements) do not further change the target state. An intuitive way to see this is that the residual state in the t register is either $|+\rangle$ or $|-\rangle$, which is invariant (up to a global phase) under CNOT. To verify this formally, one can consider an arbitrary state:

$$\left(\sum_{\mathbf{x}_{j,0}} \alpha(\mathbf{x}_{j,0}) |\mathbf{x}_{j,0}\rangle + \sum_{\mathbf{x}_{j,1}} \beta(\mathbf{x}_{j,1}) |\mathbf{x}_{j,1}\rangle \right) \otimes (|0\rangle_t + (-1)^z |1\rangle_t)$$

then applying CNOT leads to:

$$\begin{aligned}
&\left(\sum_{\mathbf{x}_{j,0}} \alpha(\mathbf{x}_{j,0}) |\mathbf{x}_{j,0}\rangle \otimes (|0\rangle_t + (-1)^z |1\rangle_t) + \sum_{\mathbf{x}_{j,1}} \beta(\mathbf{x}_{j,1}) |\mathbf{x}_{j,1}\rangle \otimes (|1\rangle_t + (-1)^z |0\rangle_t) \right) \\
&= \left(\sum_{\mathbf{x}_{j,0}} \alpha(\mathbf{x}_{j,0}) |\mathbf{x}_{j,0}\rangle + (-1)^z \sum_{\mathbf{x}_{j,1}} \beta(\mathbf{x}_{j,1}) |\mathbf{x}_{j,1}\rangle \right) \otimes (|0\rangle_t + (-1)^z |1\rangle_t)
\end{aligned}$$

which results in the two states being in tensor product. Therefore, a Hadamard basis measurement on the first subsystem has no effect on the target qubit. □

Acknowledgements

The authors thank Nico Döttling for enlightening discussions on lattice smoothing, and for sharing the proof of Lemma 5. The authors also thank Russell W.F. Lai for discussions at an early stage of this project.

P.B. and G.M. are supported by the European Research Council through an ERC Starting Grant (Grant agreement No. 101077455, ObfusQation). G.M. is also funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972.

References

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

[Bar86] David A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 1–5, 1986.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018.

[BD24] Zvika Brakerski and Nico Döttling. (Personal Communication), 2024.

[BDGM19] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Leveraging linear decryption: Rate-1 fully-homomorphic encryption and time-lock puzzles. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 407–437. Springer, Cham, December 2019.

[BDJ+24] Pedro Branco, Nico Döttling, Abhishek Jain, Giulio Malavolta, Surya Mathialagan, Spencer Peters, and Vinod Vaikuntanathan. Pseudorandom obfuscation and applications. Cryptology ePrint Archive, Paper 2024/1742, 2024.

[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 Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 533–556, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[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 Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 252–281. Springer, Cham, April 2023.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 609–629. Springer, Berlin, Heidelberg, August 2015.

[BK25] James Bartusek and Dakshita Khurana. On the power of oblivious state preparation. CRYPTO, 2025.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 868–886. Springer, Berlin, Heidelberg, August 2012.

[Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 67–95. Springer, Cham, August 2018.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Berlin, Heidelberg, August 2011.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In Moni Naor, editor, *ITCS 2014*, pages 1–12. ACM, January 2014.

[CDM21] Orestis Chardouvelis, Nico Döttling, and Giulio Malavolta. Rate-1 quantum fully homomorphic encryption. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 149–176. Springer, Cham, November 2021.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 468–497. Springer, Berlin, Heidelberg, March 2015.

[Cop96] Don Coppersmith. Finding a small root of a univariate modular equation. In Ueli Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, pages 155–165, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[DPPv22] Léo Ducas, Eamonn W. Postlethwaite, Ludo N. Pulles, and Wessel P. J. van Woerden. Hawk: Module LIP makes lattice signatures fast, compact and simple. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 65–94. Springer, Cham, December 2022.

[DSS16] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 3–32. Springer, Berlin, Heidelberg, August 2016.

[Dv22] Léo Ducas and Wessel P. J. van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 643–673. Springer, Cham, May / June 2022.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GH11] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 107–109. IEEE Computer Society Press, October 2011.

[GH19] Craig Gentry and Shai Halevi. Compressible FHE with applications to PIR. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 438–464. Springer, Cham, December 2019.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 465–482. Springer, Berlin, Heidelberg, April 2012.

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 555–564. ACM Press, June 2013.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Berlin, Heidelberg, August 2013.

[GV24] Aparna Gupte and Vinod Vaikuntanathan. How to construct quantum FHE, generically. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 246–279. Springer, Cham, August 2024.

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

[HR14] Ishay Haviv and Oded Regev. On the lattice isomorphism problem. In Chandra Chekuri, editor, *25th SODA*, pages 391–404. ACM-SIAM, January 2014.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *21st ACM STOC*, pages 12–24. ACM Press, May 1989.

[LJL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 1219–1234. ACM Press, May 2012.

[Mah22] Urmila Mahadev. Classical verification of quantum computations. *SIAM Journal on Computing*, 51(4):1172–1229, 2022.

[Mah23] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. *SIAM Journal on Computing*, 52(6):FOCS18–189–FOCS18–215, 2023.

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

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011.

[NIS] NIST post-quantum cryptography. https://csrc.nist.gov/projects/post-quantum-cryptography.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009.

[PS97] W. Plesken and B. Souvignier. Computing isometries of lattices. *J. Symb. Comput.*, 24(3–4):327–334, October 1997.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Rot11] Ron Rothblum. Homomorphic encryption: From private-key to public-key. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 219–234. Springer, Berlin, Heidelberg, March 2011.

[vGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 24–43. Springer, Berlin, Heidelberg, May / June 2010.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017.