Structural Lattice Reduction: Generalized Worst-Case to Average-Case Reductions and Homomorphic Cryptosystems

Nicolas Gama [*] Malika Izabachene [†] Phong Q. Nguyen [‡] Xiang Xie [§]

Abstract

In lattice cryptography, worst-case to average-case reductions rely on two problems: Ajtai's SIS and Regev's LWE, which both refer to a very small class of random lattices related to the group $G = \mathbb{Z}_q^n$. We generalize worst-case to average-case reductions to all integer lattices of sufficiently large determinant, by allowing G to be any (sufficiently large) finite abelian group. In particular, we obtain a partition of the set of full-rank integer lattices of large volume such that finding short vectors in a lattice chosen uniformly at random from any of the partition cells is as hard as finding short vectors in any integer lattice. Our main tool is a novel generalization of lattice reduction, which we call structural lattice reduction: given a finite abelian group G and a lattice L, it finds a short basis of some lattice $\bar{L}$ such that $L \subseteq \bar{L}$ and $\bar{L}/L \simeq G$. Our group generalizations of SIS and LWE allow us to abstract lattice cryptography, yet preserve worst-case assumptions: as an example, we provide a somewhat conceptually simpler generalization of the Alperin-Sheriff-Peikert variant of the Gentry-Sahai-Waters homomorphic scheme. We introduce homomorphic mux gates, which allows us to homomorphically evaluate any boolean function with a noise overhead proportional to the square root of its number of variables, and bootstrap the full scheme using only a linear noise overhead.

1 Introduction

A lattice is a discrete subgroup of $\mathbb{R}^m$, *e.g.* a subgroup of $\mathbb{Z}^m$. Nearly two decades after its introduction, lattice-based cryptography has emerged as a credible alternative to classical public-key cryptography based on factoring or discrete logarithm. It offers new properties (such as security based on worst-case assumptions) and new functionalities, such as noisy multilinear maps and fully-homomorphic encryption. The worst-case guarantees of lattice-based cryptography come from two major problems: the *short integer solution* (SIS) problem dating back to Ajtai's breakthrough work at STOC '96 [1], and the *learning with errors* (LWE) problem introduced by Regev at STOC '05 [38], and somewhat related to the Ajtai-Dwork cryptosystem [2]. These two average-case problems are provably as hard as solving certain lattice problems in the worst case, such as GapSVP (the decision version of the shortest vector problem in a lattice) and SIVP (finding short linearly independent lattice vectors).

The SIS problem can be defined as finding short vectors in a random lattice from a class $\mathcal{A}_{n,m,q}$ of m-dimensional integer lattices related to the finite abelian group $G = \mathbb{Z}_q^n$, where n is the dimension of the worst-case lattice problem and q needs to be sufficiently large: any $\mathbf{g} = (g_1, \ldots, g_m) \in G^m$ chosen uniformly at random defines a lattice $\mathcal{L}_\mathbf{g} \in \mathcal{A}_{n,m,q}$ formed by all $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{Z}^m$ s.t. $\sum_{i=1}^m x_i g_i = 0$ in G; and SIS asks, given $\mathbf{g}$, to find a short (nonzero) $\mathbf{x} \in \mathcal{L}_\mathbf{g}$. Here, the definition of $\mathcal{L}_\mathbf{g}$ relies on the $\mathbb{Z}$-module structure of G. The class $\mathcal{A}_{n,m,q}$ has an algebraic meaning: the distribution of $\mathcal{L}_\mathbf{g}$ turns out to be statistically close (for sufficiently large m) to the uniform distribution over the finite set $\mathcal{L}_{G,m}$ of all full-rank lattices $L \subseteq \mathbb{Z}^m$ such that $\mathbb{Z}^m/L \simeq G$. This suggests that Ajtai's lattices are very rare among all

[*] Université de Versailles and CNRS, France

[†] CEA LIST, France

[‡] INRIA, France and Tsinghua University, China

[§] Institute of Software, Chinese Academy of Sciences, China integer lattices: in fact, Nguyen and Shparlinski [31] recently showed that the set $\cup_{G \text{ cyclic}}\mathcal{L}_{G,m}$ of all full-rank integer lattices $L \subseteq \mathbb{Z}^m$ such that $\mathbb{Z}^m/L$ is cyclic (unlike $\mathbb{Z}_q^n$) has natural density $1/[\zeta(6)\prod_{k=4}^m \zeta(k)] \approx 85\%$ (for large m), which implies that Ajtai's classes $\mathcal{A}_{n,m,q}$ form a minority of lattices among all integer lattices.

This motivates the natural question of whether other classes of random lattices enjoy similar worst-case to average-case reductions: in particular, if we call GSIS the generalization of SIS to any finite abelian group G, does GSIS have similar properties as SIS for other groups than $G = \mathbb{Z}_q^n$? This would imply that the random lattices of the class $\mathcal{L}_{G,m}$ are also hard. Ajtai (in the proceedings version of [1]) and later Regev [37] noticed that the choice $G = \prod_{i=1}^n \mathbb{Z}_{q_i}$ where the q_i's are distinct prime numbers of similar bit-length also worked. Micciancio [25] gave another special choice of G: his G is actually constructed by an algorithm [25, Lemma 2.11] given as input a very special lattice (for which solving the closest vector problem is easy); if the input lattice is $\mathbb{Z}^n$, then $G = (\mathbb{Z}_q)^n$. However, all these choices of G are arguably very special, and it was unclear if the hardness properties held outside a small family of finite abelian groups.

A similar question can be asked for LWE, which is known as a dual problem of SIS, and has been used extensively in lattice-based encryption. However, in order to define GLWE by analogy with GSIS, we need to change the usual definition of LWE based on linear algebra. Any finite abelian group G is isomorphic to its dual group $\hat{G}$ formed by its characters, *i.e.* homomorphisms from G to the torus $\mathbb{T} = \mathbb{R}/\mathbb{Z}$. We define search-GLWE as the problem of learning a character $\hat{s} \in \hat{G}$ chosen uniformly at random, given noisy evaluations of $\hat{s}$ at (public) random points $g_1, \ldots, g_m \in G$, namely one is given g_i and a "Gaussian" perturbation of $\hat{s}(g_i)$ for all $1 \leq i \leq m$. Decisional-GLWE is defined as the problem of distinguishing the previous "Gaussian" perturbations of $\hat{s}(g_i)$ from random elements in $\mathbb{T}$. If $G = (\mathbb{Z}_q)^n$, it can be checked that GLWE is LWE. If $G = \mathbb{Z}_p$ for some large prime p, search-GLWE is a randomized version of Boneh-Venkatesan's *Hidden Number Problem* [9] (introduced to study the bit-security of Diffie-Hellman key exchange, but also used in side-channel attacks on discrete-log based signatures [30]), which asks to recover a secret number $s \in \mathbb{Z}_p$, given random $t_1, \ldots, t_m$ chosen uniformly from $\mathbb{Z}_p$ and approximations of each $st_i \bmod p$. Here, randomized means that the approximations given are "Gaussian" perturbations of $st_i \bmod p$. Thus, GLWE captures LWE and the Hidden Number Problem as a single problem, instantiated with different groups. Alternatively, GLWE can be viewed as a lattice problem: solving a randomized version of bounded distance decoding (with "Gaussian" errors) for the dual lattice of $\mathcal{L}_{\mathbf{g}}$.

Our results. We show that the worst-case to average-case reductions for SIS and LWE (search and decisional) can be generalized to GSIS and GLWE, provided that G is any sufficiently large finite abelian group, *e.g.* of order $n^{\Omega(\max(n,\mathrm{rank}(G)))}$ if n is the dimension of the worst-case lattice problem and rank (G) denotes the minimal size of a generating set for G. For GSIS and search-GLWE, our reductions are direct from worst-case lattice problems. And we transfer all decisional-LWE hardness results to decisional-GLWE, by reducing decisional-LWE to decisional-GLWE (under similar size constraints on G): we do so by generalizing the modulus-dimension switching technique of Brakerski *et al.* [12].

Our reductions are based on a new tool, which we call structural lattice reduction, and which might be of independent interest: Becker *et al.* [8] recently used it to design new exponential-space algorithms for lattice problems. In lattice reduction, one is given a full-rank lattice $L \subseteq \mathbb{Z}^n$ and wants to find a short basis of L. In our structural lattice reduction, one is further given a finite abelian group G of rank $\leq n$, and wants to find a short basis of some overlattice $\bar{L}$ of L such that $\bar{L}/L \simeq G$ effectively, *i.e.* the map φ in the short exact sequence $0 \to L \xrightarrow{\mathrm{id}} \bar{L} \xrightarrow{\varphi} G \to 0$ is efficiently computable. Our key point is that previous worst-case to average-case reductions (*e.g.* [20, 12]) implicitly used a trivial case of structural lattice reduction: if B is a short basis of a full-rank lattice $L \subseteq \mathbb{Z}^n$ and q is an integer, then $q^{-1}B$ is a short basis of the lattice $\bar{L} = q^{-1}L$

such that $\bar{L}/L \simeq \mathbb{Z}_q^n$, which explains the importance of $\mathbb{Z}_q^n$ in SIS and LWE.

Our GSIS reduction shows that in some sense all integer lattices are hard. Indeed, the set of full-rank lattices $L \subseteq \mathbb{Z}^m$ (of sufficiently large co-volume $\geq n^{\Omega(m)}$) can be partitioned based on the finite abelian group $\mathbb{Z}^m/L$, and the reduction implies that each partition cell $\mathcal{L}_{G,m}$ has this worst-case to average-case property: finding short vectors in a lattice chosen uniformly at random from $\mathcal{L}_{G,m}$ is as hard as finding short vectors in any integer lattice of dimension n.

Consider the special case $G = \mathbb{Z}_p$ for a large prime p. Then our GSIS reduction provides the first hardness results for the random lattices in $\mathcal{L}_{\mathbb{Z}_p,m}$ used in many experiments [18, 14] to benchmark lattice reduction algorithms, as well as in Darmstadt's SVP internet challenges. And our GLWE reduction provides a general hardness result for the hidden number problem: previously, [12, Cor 3.4] established the hardness for the hidden number problem when the large prime p is replaced by q^n where q is smooth.

Finally, our generalizations of SIS and LWE allow us to abstract (the many) lattice-based schemes based on SIS and/or LWE, where the role of $G = (\mathbb{Z}_q)^n$ was not very explicit in most descriptions (typically based on linear algebra). We believe such an abstraction can have several benefits. First, it can clarify analyses and designs: the El Gamal cryptosystem is arguably better described with an arbitrary group G, rather than by focusing on the historical choice $G = \mathbb{Z}_p^*$; comparisons and analogies with "traditional" public-key cryptography based on factoring or discrete logarithm will be easier. We illustrate this point by providing a somewhat conceptually simpler GLWE-based generalization of the Alperin-Sheriff-Peikert variant [3] of the Gentry-Sahai-Waters homomorphic scheme [21]: this generalization becomes essentially as simple as trapdoor-based fully-homormophic encryption proposals such as [39]. It is based on a GLWE variant of El Gamal encryption, which naturally generalizes Regev's LWE encryption [38]. We also provide a new decryption circuit based on Mux gates, which can bootstrap the system with a polynomial noise overhead, and is arguably simpler than [3]. Second, it opens up the possibility of obtaining more efficient schemes using different choices of G than $G = (\mathbb{Z}_q)^n$. We do not claim that there are better choices than $G = (\mathbb{Z}_q)^n$, but such a topic is worth investigating, which we leave to future work. Many factors influence efficiency: trapdoor generation, hashing, efficiency of the security reduction, *etc.* For instance, hashing onto $\mathbb{Z}_p$ can sometimes be more efficient than onto $(\mathbb{Z}_q)^n$ for large n, which could be useful in certain settings, like digital signatures.

Related work. Baumslag *et al.* also introduced in [7] group generalizations of LWE, targeting non-commutative groups, but did not obtain any hardness result. A follow-up in [17] only showed a self-reducibility property on the problem for some special non-commutative groups.

Open problems. The recent reduction of Brakerski *et al.* [12] proves the hardness of decisional-LWE for a wide range of parameters, without establishing a direct search-to-decision equivalence for all these parameters. Similarly, our strongest hardness result for decisional-GLWE bypasses the one for search-GLWE. It is unknown if there is a direct search-to-decision equivalence for GLWE over all sufficiently large finite abelian groups. It would also be interesting to see if structural lattice reduction can be adapted to the ring setting, to obtain more hardness results based on worst-case assumptions for ideal lattices. Finally, our GSIS/GLWE reductions require the order of G to be sufficiently large compared to the worst-case lattice dimension, and it is interesting to reduce as much as possible this constraint: in particular, the case $G = \mathbb{Z}_2^n$ for GLWE corresponds essentially to LPN, whose hardness is well-known to be open; here, the order 2^n does not grow quickly enough with respect to the rank n to be covered by our reduction. On the other hand, Micciancio and Peikert [27] recently showed how to decrease q for SIS.

Roadmap. The paper is organized as follows. In Sect. 2, we recall background on lattices. In Sect. 3, we discuss factor groups of integer lattices, and introduce our group generalizations of SIS and LWE. In Sect. 4, we introduce structural lattice reduction, which will be used in all our reductions. We show hardness of GSIS in Sect. 5 by generalizing the SIS reductions. In Sect. 6, we show hardness of decisional-GLWE. In Sect. 7, we give an example of abstracting lattice cryptography: El Gamal-like encryption and fully-homomorphic encryption from GLWE. Detailed missing proofs can be found in appendix. In App. E, we generalize the classical search-LWE hardness result to search-GLWE. In App. H, we compare precisely our structural reduction with previous work of Ajtai [1] and Micciancio [25]. In App. I, we discuss the relationships between GSIS, GLWE and HNP, and compare with previous work of Brakerski *et al.* [12].

2 Background and Notation

$\mathbb{Z}_q$ denotes $\mathbb{Z}/q\mathbb{Z}$. We use row notation for vectors and matrices. I_n denotes the $n \times n$ identity matrix. A function $\mathrm{negl}(n)$ is *negligible*, if it vanishes faster than the inverse of any polynomial in n. For an $n \times m$ matrix B, $\|B\| = \max_{1 \leq i \leq n} \|\mathbf{b}_i\|$ denotes the norm of its longest row vector.

Lattices. A *lattice* L is a discrete subgroup of $\mathbb{R}^m$: it is of the form $L(B) = \{\sum_{i=1}^{n} \alpha_i \mathbf{b}_i,\ \alpha_i \in \mathbb{Z}\}$ for some set $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of linearly independent vectors, called a *basis*. When $L \subseteq \mathbb{Z}^m$, L is an *integer lattice*. The dimension n of $\mathsf{span}(L)$ is the *dimension* $\mathsf{dim}(L)$ of L. The *(co)-volume* $\mathsf{vol}(L)$ is $\sqrt{\det(BB^t)}$ for any basis B of L. For $1 \leq i \leq \mathsf{dim}(L)$, $\lambda_i(L)$ is the i-th minimum of L, *i.e.* the smallest radius of the 0-centered ball containing at least i linearly independent lattice vectors. The *dual lattice* $L^\times$ is the set of all $\mathbf{u} \in \mathsf{span}(L)$ s.t. $\langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{Z}$ for all $\mathbf{v} \in L$. If B is a basis of L, its *dual basis* $B^\times = (BB^t)^{-1}B$ is a basis of $L^\times$. For a factor $\gamma = \gamma(n) \geq 1$, GapSVP_γ asks, given $d \in \mathbb{R}_{\geq 0}$ and a basis B of an n-dim lattice L, to decide if $\lambda_1(L) \leq d$ or $\lambda_1(L) > \gamma d$. $\mathrm{ApproxSIVP}_\gamma$ asks a full-rank family of lattice vectors of norm $\leq \gamma \lambda_n(L)$.

Gram-Schmidt Orthogonalization (GSO). Let $B = (\mathbf{b}_1, ..., \mathbf{b}_n)$ be a lattice basis. The GSO of B is the unique decomposition $B = \mu \cdot D \cdot Q$, where μ is a lower triangular matrix with unit diagonal, D is a positive diagonal matrix, and Q has orthonormal rows. We let $B^* = DQ$ whose i-th row $\mathbf{b}_i^*$ is $\pi_i(\mathbf{b}_i)$, where π_i denotes the orthogonal projection of $\mathbf{b}_i$ over $\mathsf{span}\{\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}\}^\perp$. We use the notation $B_{[i,j]}$ for the block $[\pi_i(\mathbf{b}_i), \ldots, \pi_i(\mathbf{b}_j)]$. If $B^\times$ is the dual basis of B and $(B^\times)^*$ denotes its GSO matrix, then $\|(\mathbf{b}_i^\times)^*\| \cdot \|\mathbf{b}_{n-i+1}^*\| = 1$ for $1 \leq i \leq n$.

(Explicit) Finite abelian groups. Any finite abelian group G is isomorphic to a product $\prod_{i=1}^{k} \mathbb{Z}_{q_i}$ of cyclic groups. We call *rank* of G the minimal number of cyclic groups in such decompositions: this should not be confused with the rank of an abelian group, which would be zero here. We say that G is *explicit* if one knows integers $q_1, \ldots, q_k$ and an isomorphism $\prod_{i=1}^{k} \mathbb{Z}_{q_i} \to G$ computable in polynomial time: we will assume that k is the rank and each q_{i+1} divides q_i, because from an arbitrary decomposition, one can always derive the rank and such q_i's in polynomial time. The isomorphism induces k generators $e_1, \ldots, e_k \in G$ s.t. $G = \langle e_1 \rangle \oplus \cdots \oplus \langle e_k \rangle$ and each e_i has order q_i. If the inverse of the isomorphism is also computable in polynomial time, we say that G is *fully-explicit*.

Overlattices and exact sequences. When a lattice $\bar{L}$ contains a sublattice L of the same dimension n, $\bar{L}$ is an *overlattice* of L. Then $\bar{L}/L$ is a finite abelian group of rank $\leq n$ and order $\mathrm{vol}(L)/\mathrm{vol}(\bar{L})$. Then $0 \to L \overset{\mathrm{id}}{\to} \bar{L} \overset{\varphi}{\to} G \to 0$ is a short exact sequence for some φ, *i.e.* $\varphi : \bar{L} \to G$ is a surjective morphism s.t $\ker \varphi = L$. In other words, φ represents the isomorphism $\bar{L}/L \simeq G$.

Lattice reduction. Gentry *et al.* [20] introduced the *basis length* of a lattice L as $\mathrm{bl}(L) = \min_{\mathsf{basis}\ B} \|B^*\|$. Then: $\lambda_n(L) \geq \mathrm{bl}(L) \geq \lambda_n(L)/\sqrt{n}$, $\mathrm{bl}(L) \geq \lambda_1(L)$, and $\mathrm{bl}(L) \geq \mathrm{vol}(L)^{1/n}$.

Lattice reduction can find bases B with small $\|B^*\|$. For instance, a basis B is LLL-reduced [23] with factor $\varepsilon_{\text{LLL}} \geq 0$ if its GSO satisfies $|\mu_{i,j}| \leq \frac{1}{2}$ for all $1 \leq j < i$ and each block $B_{[i,i+1]}$ satisfies: $\|\mathbf{b}_i^*\|^2 \leq (1+\varepsilon_{\text{LLL}})(\left\|\mathbf{b}_{i+1}^*\right\|^2 + \mu_{i+1,i}\,\|\mathbf{b}_i^*\|^2)$. Then it is folklore that: $\|B^*\| \leq \left((1+\varepsilon_{\text{LLL}})\sqrt{4/3}\right)^{(n-1)/2} \text{bl}(L)$. Given as input $\varepsilon_{\text{LLL}} > 0$ and a basis B of a lattice $L \subseteq \mathbb{Z}^n$, the LLL algorithm [23] outputs an LLL-reduced basis of factor ε_{LLL} in time polynomial in $1/\varepsilon_{\text{LLL}}$ and size(B). Usually, one selects ε_{LLL} s.t. $(1+\varepsilon_{\text{LLL}})\sqrt{4/3} = \sqrt{2}$ or $\varepsilon_{\text{LLL}} = 1/\operatorname{poly}(n)$.

2.1 Gaussian Measures

The statistical distance between two distributions $\mathcal{P}$ and $\mathcal{Q}$ over a domain X is $\Delta(\mathcal{P},\mathcal{Q}) = \frac{1}{2}\int_{a\in X}|\mathcal{P}(a)-\mathcal{Q}(a)|da$ or $\frac{1}{2}\sum_{a\in X}|\mathcal{P}(a)-\mathcal{Q}(a)|$ when X is discrete. Two distributions $\mathcal{P}$ and $\mathcal{Q}$ are (statistically) ϵ-indistinguishable if $\Delta(\mathcal{P},\mathcal{Q}) < \epsilon$. We write $\mathbf{y} \leftarrow \mathcal{P}$ (resp. $\leftarrow_\varepsilon \mathcal{P}$) to denote a sample from the distribution $\mathcal{P}$ (resp. a distribution ϵ-indistinguishable from $\mathcal{P}$). And the symbol $\leftarrow_\approx$ means $\leftarrow_\varepsilon$ for some negligible function ε.

Gaussian Distributions. The *Gaussian Distribution* (over $\mathbb{R}^n$) $\mathcal{D}_{\mathbb{R}^n,\sigma,\mathbf{c}}$ centered at $\mathbf{c} \in \mathbb{R}^n$ of parameter $\sigma \in \mathbb{R}_{\geq 0}$ is defined by a density function proportional to $\rho_{\mathbb{R}^n,\sigma,\mathbf{c}}(\mathbf{x}) = \exp\left(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2\right)$. If $\mathbf{c}$ is omitted, then $\mathbf{c} = 0$. For any countable subset $C \subseteq \mathbb{R}^n$ (*e.g.* a lattice L or a coset $\mathbf{x}+L$), $\rho_{\mathbb{R}^n,\sigma,\mathbf{c}}(C)$ denotes $\sum_{\mathbf{u}\in C}\rho_{\mathbb{R}^n,\sigma,\mathbf{c}}(\mathbf{u})$. The *discrete Gaussian distribution* $\mathcal{D}_{C,\sigma,\mathbf{c}}$ over a lattice or a coset $C \subset \mathbb{R}^n$ is defined by $\mathcal{D}_{C,\sigma,\mathbf{c}}(\mathbf{x}) = \rho_{\mathbb{R}^n,\sigma,\mathbf{c}}(\mathbf{x})/\rho_{\mathbb{R}^n,\sigma,\mathbf{c}}(C)$ where $\mathbf{x} \in C$. It is known how to sample efficiently the discrete Gaussian distribution over lattices to within negligible distance [20, 35], or even exactly [12]:

Lemma 2.1 *There is a polynomial-time algorithm which, given $\mathbf{c} \in \mathbb{Q}^n$, a basis B of a lattice $L \subseteq \mathbb{Q}^n$ and a parameter $\sigma \geq \|B^*\| \cdot \sqrt{\ln(2n+4)/\pi}$, outputs a sample with distribution $\mathcal{D}_{L,\sigma,\mathbf{c}}$.*

Reciprocally, on can construct a short lattice basis from short discrete Gaussian samples:

Proposition 2.2 *(Cor. of [37, Lemma 14]) Let $\varepsilon > 0$ and $L(B)$ be an n-dimensional lattice. Given a set of $m = O(n)$ independent Gaussian samples $(\mathbf{y}_i \leftarrow_\varepsilon \mathcal{D}_{L,s_i})$ s.t. $\sqrt{2}\eta_\varepsilon(L) \leq s_i \leq \sigma$, $1 \leq i \leq m$, one can compute in polynomial time a basis C of L s.t. $\|C^*\| \leq \sqrt{n/2\pi} \cdot \max_i s_i$.*

Modular Distributions and Smoothing Parameter. The continuous distribution $\mathcal{D}_{\mathbb{R}^n,\sigma,c}$ and discrete distribution $\mathcal{D}_{\bar{L},\sigma,c}$ over an overlattice $\bar{L} \supseteq L$ can be projected modulo L. Thus $\mathcal{D}_{\mathbb{R}^n/L,\sigma,c}$ (resp. $\mathcal{D}_{\bar{L}/L,\sigma,c}$) has a density function $\mathcal{D}_{\mathbb{R}^n,\sigma,\mathbf{c}}(\mathbf{x}+L)$ for $\mathbf{x} \in \mathbb{R}^n/L$ (resp. $\bar{L}/L$). Both $\mathcal{D}_{\mathbb{R}^n/L,\sigma}$ and $\mathcal{D}_{\bar{L}/L,\sigma}$ converge (uniformly) to the uniform distribution when σ increases. This is quantified by the *smoothing parameter* $\eta_\varepsilon(L)$ (where $\varepsilon > 0$) introduced by Micciancio and Regev [28] as the minimal $\sigma > 0$ s.t. $\rho_{\mathbb{R}^n,\frac{1}{\sigma}}(L^\times\setminus\{0\}) \leq \varepsilon$, *i.e.* $\left\|\mathcal{D}_{\mathbb{R}^n/L,\sigma}(\mathbf{x}+L) - \frac{1}{\text{vol}(L)}\right\|_\infty \leq \frac{\varepsilon}{\text{vol}(L)}$ by Poisson's summation formula, which proves:

Lemma 2.3 (see Cor 2.8 of [20]) *If $\bar{L}$ is an overlattice of L, $\varepsilon \in (0,1/2)$, $\sigma \geq \eta_\varepsilon(L)$ and $\mathbf{c} \in \mathbb{R}^n$, then $\mathcal{D}_{\bar{L}/L,\sigma,\mathbf{c}+L}$ is within stat. distance $\leq 2\varepsilon$ from the uniform distribution over $\bar{L}/L$.*

For any n-dim basis B, $\eta_\varepsilon(L(B)) \leq \eta_\varepsilon(L(B^*)) \leq \eta_\varepsilon(\mathbb{Z}^n) \cdot \|B^*\|$ where $\eta_\varepsilon(\mathbb{Z}^n) \leq \sqrt{\log\left(2n\cdot(1+\frac{1}{\varepsilon})\right)/\pi}$. In particular, $\eta_\varepsilon(L) \leq \eta_\varepsilon(\mathbb{Z}^n)\cdot\text{bl}(L)$. Finally, we give a technical lemma on the dot product of a discrete Gaussian (proved in App. A.2), analogous to [38, 35].

Lemma 2.4 (Dot product convolution) *Let $\mathbb{K} = \mathbb{R}$ or $\mathbb{T}$. Let $c \in \mathbb{R}$, $\mathbf{u} \in \mathbb{R}^n$, $\alpha,\sigma \in \mathbb{R}_{\geq 0}$, $\varepsilon \in (0,1/2)$ and $\mathbf{z}+L$ be a coset of an n-dim lattice $L \subseteq \mathbb{R}^n$. Assume that*

$\left(\frac{1}{\sigma^2}+\frac{\|\mathbf{u}\|^2}{\alpha^2}\right)^{-1/2} \geq \eta_\varepsilon(L)$. *Then* $\mathcal{D}_{\mathbb{K},\alpha,c+\langle\mathbf{u},\mathbf{v}\rangle}$ *where* $\mathbf{v} \leftarrow \mathcal{D}_{\mathbf{z}+L,\sigma}$ *is within statistical distance* $\leq 4\varepsilon$ *from* $\mathcal{D}_{\mathbb{K},\sqrt{\alpha^2+\sigma^2\|\mathbf{u}\|^2},c}$. *This still holds when* $\mathbb{K}=\frac{1}{N}\mathbb{Z}$ *or* $\frac{1}{N}\mathbb{Z}/\mathbb{Z}$ *if* $\alpha \geq \eta_\varepsilon(\frac{1}{N}\mathbb{Z})$.

2.2 SIS and LWE

Let $G=\mathbb{Z}_q^n$. SIS(m,n,q,β) [1] asks, given $\mathbf{g}=(g_1,\ldots,g_m)\in_R G^m$, to find a non-zero vector $\mathbf{x}\in\mathbb{Z}^m$ s.t. $\sum_{i=1}^m x_i g_i = 0$ and $\|\mathbf{x}\|\leq\beta$. Such an $\mathbf{x}$ exists if $\beta\geq\sqrt{m}q^{n/m}$. Ajtai [1] proved that SIS (with suitable parameters) is at least as hard as approximating SIVP in the worst case for dimension n to within some polynomial factor: in the best reduction known [20], the factor is $\tilde{O}(n)$. In LWE(m,n,q,β) [38], one picks $\mathbf{s}\in_R G$ and $(g_1,\ldots,g_m)\in_R G^m$. Let A be the $m\times n$ matrix whose i-th row is g_i. LWE asks to recover $\mathbf{s}\in G$, given as input $(A,\mathbf{s}A^t+\mathbf{e})$ where $\mathbf{e}\in\mathbb{Z}_q^m$ is chosen with distribution $\mathcal{D}_{\mathbb{Z}^m,\beta q}$ (as in [35]) or the original distribution of [38].

3 Lattice Factor Groups and Generalizations of SIS and LWE

In this section, we present our group generalizations of SIS and LWE, which are related to factor groups of integer lattices.

3.1 Lattice Factor Groups

If L is a full-rank lattice $\subseteq\mathbb{Z}^m$, its factor group $\mathbb{Z}^m/L$ is a finite abelian group of order vol(L). For any finite abelian group G, denote by $\mathcal{L}_{G,m}$ the (finite) set of full-rank lattices $L\subseteq\mathbb{Z}^m$ such that $\mathbb{Z}^m/L\simeq G$. The following elementary characterization of $\mathcal{L}_{G,m}$ is a consequence of [33]:

Theorem 3.1 *Let G be a finite abelian group and L be a full-rank lattice in $\mathbb{Z}^m$. Then $L\in\mathcal{L}_{G,m}$ if and only if G has rank $\leq m$ and there exists $\mathbf{g}=(g_1,\ldots,g_m)\in G^m$ such that the g_i's generate G and $L=\mathcal{L}_\mathbf{g}$ where* $\mathcal{L}_\mathbf{g}=\{(x_1,\ldots,x_m)\in\mathbb{Z}^m \text{ s.t. } \sum_{i=1}^m x_i g_i = 0 \text{ in } G\}$.

Given G, Alg. 1 shows how to sample efficiently lattices from the uniform distribution over $\mathcal{L}_{G,m}$, and its correctness follows from (the trivial) Lemma 3.2. Previously, efficient sampling was only known for $G=\mathbb{Z}_p$ where p is a large prime (see [22]).

Algorithm 1 Sampling lattices of given factor group

```
Input: Integer $m \geq 1$ and a finite abelian group $G = \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_k}$ such that $1 \leq k \leq m$.
Output: A random lattice from the uniform distribution over $\mathcal{L}_{G,m}$.
1: Generate elements $g_1,\ldots,g_m$ uniformly at random from $G$ until the $g_i$'s generate $G$.
2: Return the lattice $\mathcal{L}_\mathbf{g}$ where $\mathbf{g}=(g_1,\ldots,g_m)\in G^m$.
```

Lemma 3.2 *Let G be a finite abelian group. Let $\mathbf{g}=(g_1,\ldots,g_m)\in G^m$ be such that the g_i's generate G. Let $\mathbf{h}=(h_1,\ldots,h_m)\in G^m$. Then $\mathcal{L}_\mathbf{g}=\mathcal{L}_\mathbf{h}$ if and only if there is an automorphism ψ of G such that $h_i=\psi(g_i)$ for all $1\leq i\leq m$. In such a case, ψ is uniquely determined.*

We note that several implementations of lattice-based cryptography (such as [19]) implicitly used lattices in $\mathcal{L}_{G,m}$ for some large cyclic group G. Recently, Nguyen and Shparlinski [31] showed that such lattices are dominant: the set $\cup_{G \text{ cyclic}}\mathcal{L}_{G,m}$ of all full-rank integer lattices $L\subseteq\mathbb{Z}^m$ such that $\mathbb{Z}^m/L$ is cyclic has natural density $1/[\zeta(6)\prod_{k=4}^m\zeta(k)]\approx 85\%$ (for large m).

3.2 The Group-SIS Problem (GSIS)

We introduce the *Group-SIS* problem (GSIS), which is a natural generalization of SIS to arbitrary finite abelian groups. The GSIS parameters are $m \geq 1$, a finite abelian group G and a bound $\beta \in \mathbb{R}_{\geq 0}$. One picks a sequence $\mathbf{g} = (g_1, \ldots, g_m) \in G^m$ uniformly at random. $\mathrm{GSIS}(G, m, \beta)$ asks to find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ s.t. $\sum_{i=1}^{m} x_i g_i = 0$ and $\|\mathbf{x}\| \leq \beta$. In other words, GSIS asks to find short vectors in random relation lattices $\mathcal{L}_{\mathbf{g}} = \{\mathbf{x} \in \mathbb{Z}^m \text{s.t.} \sum_{i=1}^{m} x_i g_i = 0\}$. For instance, $\mathrm{GSIS}(\mathbb{Z}_q^n, m, \beta)$ is SIS, and $\mathrm{GSIS}(\mathbb{Z}_q, m, \beta)$ is finding short vectors in random m-dimensional co-cyclic lattices of volume q. If $\#G$ denotes the order of G, the existence of a GSIS-solution is guaranteed if $\beta \geq \sqrt{m}(\#G)^{1/m}$.

GSIS is connected to $\mathcal{L}_{G,m}$ as follows. It is known [24, 32] that as soon as $m \geq n + 2\log\log\#G + 5$ (resp. $m > 2\log\#G + 2$), $g_1, \ldots, g_m$ generate the whole group G with probability $\geq 1/e$ (resp. $\geq 1 - 1/\#G$), in which case $\mathbb{Z}^m/\mathcal{L}_{\mathbf{g}} \simeq G$. In particular, if $m > 2\log\#G + 2$, then the distribution of GSIS lattices $\mathcal{L}_{\mathbf{g}}$ is statistically close to the uniform distribution over $\mathcal{L}_{G,m}$, because it is statistically close to the distribution produced by Alg. 1, in which case, solving GSIS is equivalent to finding short vectors in random lattices from $\mathcal{L}_{G,m}$.

Finally, we note that to establish hardness of GSIS, it suffices to focus on low-rank groups G. Indeed, if $G' = G \times H$ for some finite abelian group G, H, then GSIS over G can trivially be reduced to GSIS over G', by "projecting" G' to G.

3.3 The Group-LWE Problem (GLWE)

We introduce the *Group-LWE* problem (GLWE), which generalizes LWE. It uses the torus $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ and a finite abelian group G. Let $\hat{G}$ be the dual group of homomorphisms from G to $\mathbb{T}$: it is isomorphic to G but not canonically. If G is explicit, $G = \oplus_{i=1}^{k} \langle e_i \rangle$ where e_i has order q_i, and $\hat{G}$ is generated by $\hat{e}_1, \ldots, \hat{e}_k$ defined as $\hat{e}_i(\sum_{j=1}^{k} \alpha_j e_j) = \alpha_i/q_i \mod 1$ where $0 \leq \alpha_j < q_j$.

Let $\mathcal{S}$ be a known distribution over $\hat{G}$. Search-GLWE is the problem of learning a character $\hat{s} \in \hat{G}$ picked from $\mathcal{S}$, given noisy evaluations of $\hat{s}$ at (public) random points $a_1, \ldots, a_m \in G$, namely one is given (for all i's) a_i and a "Gaussian" perturbation of $\hat{s}(a_i)$. Like LWE, several noise distributions are possible. As in [38], we focus on the continuous distribution where $\hat{s}(a)$ is shifted by an error $e \leftarrow \mathcal{D}_{\mathbb{R},\alpha}$. These distributions need to be discretized in order to have a finite representation. In App. B.4, we present discrete versions of GLWE and show that they are at least as hard as the continuous version for some suitable parameters, which explains why we only consider the continuous GLWE problem in the rest of the article:

Definition 3.3 *Let G be an explicit finite abelian group: $G = \oplus_{i=1}^{k} \langle e_i \rangle$. Let $\alpha > 0$ and $\hat{s} \in \hat{G}$.*

- *$A_{G,\alpha}(\hat{s})$ is the distribution over $G \times \mathbb{T}$ defined by choosing $a \in G$ uniformly at random, setting $b \leftarrow \mathcal{D}_{\mathbb{T},\alpha,\hat{s}(a)}$, and outputting $(a, b) \in G \times \mathbb{T}$.*
- *Search-*$\mathrm{GLWE}_{G,\alpha}(\mathcal{S})$ *asks to find $\hat{s}$ from $A_{G,\alpha}(\hat{s})$ for a fixed $\hat{s} \to \mathcal{S}$ given arbitrarily many independent samples. By finding $\hat{s}$, we mean finding $s_i \in \mathbb{Z}$ s.t. $\hat{s} = \sum_{i=1}^{k} s_i \hat{e}_i$.*
- *Decisional-*$\mathrm{GLWE}_{G,\alpha}(\mathcal{S})$ *asks to distinguish $A_{G,\alpha}(\hat{s})$ from the uniform distribution over $G \times \mathbb{T}$ for a fixed $\hat{s}$ sampled from $\mathcal{S}$ given arbitrarily many independent samples.*
- *For $0 < \alpha < 1$, (Search) Decisional-*$\mathrm{GLWE}_{G,\leq\alpha}(\mathcal{S})$ *is the problem of solving (Search) Decisional-*$\mathrm{GLWE}_{G,\beta}(\mathcal{S})$ *for any $\beta \leq \alpha$ respectively,* i.e. *when the noise parameter is unknown yet $\leq \alpha$, by analogy with LWE.*

Search-$\mathrm{GLWE}_{G,m,\alpha}(\mathcal{S})$ *and Decisional-*$\mathrm{GLWE}_{G,m,\alpha}(\mathcal{S})$ *denote the variants where the algorithms have a bounded number m of samples. If $\mathcal{S}$ is omitted, it is the uniform distribution over $\hat{G}$.*

If $G = \mathbb{Z}_q^n$, the canonical representation of G and $\hat{G}$ shows that GLWE is equivalent to the fractional version of Regev's original LWE. If $G = \mathbb{Z}_p$ for some prime p, then $\hat{G}$ can be defined by multiplications: $\hat{\mathbf{s}}$ is the homomorphism mapping any $t \in \mathbb{Z}_p$ to $ts/p \mod 1$. Thus, GLWE can be viewed as a randomized version of Boneh-Venkatesan's *Hidden Number Problem* [9]: recover a secret number s mod p, given approximations of st_i mod p for many random integers t_i's. By analogy with LWE (see [38, 12]), there is a folklore reduction from (Search) Decisional-$\mathrm{GLWE}_{G,\leq\alpha}(\mathcal{S})$ to (Search) Decisional-$\mathrm{GLWE}_{G,\alpha}(\mathcal{S})$, respectively.

Lemma 3.4 *(Adapted from [12, Lemma 2.13]) Let $\mathcal{A}$ be an algorithm for Decisional-*$\mathrm{GLWE}_{G,m,\alpha}(\mathcal{S})$ *(resp. Search) with advantage at least $\varepsilon > 0$. Then there exists an algorithm $\mathcal{B}$ for Decisional-*$\mathrm{GLWE}_{G,m',\leq\alpha}(\mathcal{S})$ *(resp. Search) using oracle access to $\mathcal{A}$ and with advantage $\geq 1/3$, where both m' and its running time are poly*$(m, 1/\varepsilon, \log \#G)$.

Proof. (Sketch, see App. B.3 for a detailed proof). Like in LWE, the basic idea is to add noises in small increments to the distribution obtained from the challenger, and feed it to the oracle solving the Decisional-$\mathrm{GLWE}_{G,\alpha}(\mathcal{S})$ (resp. Search) and estimate the behavior of the oracle. □

4 Structural Lattice Reduction

4.1 Overview

A basic result (following from structure theorems of finitely-generated modules over principal ideal domains) states that for any full-rank sublattice L of a full-rank lattice $\bar{L} \subseteq \mathbb{R}^n$, there exists a basis $\bar{B} = (\bar{\mathbf{b}}_1, \ldots, \bar{\mathbf{b}}_n)$ of $\bar{L}$ and integers $q_1, \ldots, q_n \geq 1$ such that $q_1 \geq q_2 \geq \cdots \geq q_n \geq 1$ and $B = (q_1\bar{\mathbf{b}}_1, \ldots, q_n\bar{\mathbf{b}}_n)$ is a basis of L. The q_i's can be made unique by selecting them as powers of prime numbers, or by requiring each q_{i+1} to divide q_i, in which case $q_1, \ldots, q_n$ are the *elementary divisors* of the pair $(\bar{L}, L)$: for instance, if $\bar{L} = \mathbb{Z}^n$ and L is a full-rank integer lattice, the q_i's are the diagonal coefficients of the Smith normal form of L.

In this section, we introduce a lattice reduction converse, which we call *structural lattice reduction.* Lattice reduction asks to find a short basis of a given full-rank lattice $L \subseteq \mathbb{Z}^n$. In structural lattice reduction, one is further given a finite abelian group G of rank $\leq n$, and wants to find a *short* basis of some overlattice $\bar{L}$ of L such that $\bar{L}/L \simeq G$ effectively. More precisely, given a basis B of a full-rank lattice $L \subseteq \mathbb{Z}^n$, a suitable bound $\sigma > 0$ and integers $q_1 \geq \cdots \geq q_k$ defining $G = \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_k}$, one asks to compute a basis $\bar{B}$ of an overlattice $\bar{L} \supseteq L$ such that $\|\bar{B}^*\| \leq \sigma$ and $B = (q_1\bar{\mathbf{b}}_1, \ldots, q_k\bar{\mathbf{b}}_k, \bar{\mathbf{b}}_{k+1}, \ldots, \bar{\mathbf{b}}_n)$ is a basis of L. Interestingly, we do not require the input basis B to have integer or rational coefficients, as long as its Gram-Schmidt coefficients are known with enough precision. Indeed, our structural reduction algorithm can simply focus on finding the rational transformation matrix between $\bar{B}$ and B.

Previous worst-case to average-case reductions implicitly used the homocyclic group $G = \mathbb{Z}_q^n$, thus $\bar{L} = L/q$. Here, finding a basis $\bar{B}$ of $\bar{L}$ with small $\|\bar{B}^*\|$ is the same as finding the basis $B = q\bar{B}$ of L with small $\|B^*\|$, which is just classical lattice reduction. However, we obtain new problems and applications by considering different choices of G. In the trivial case $G = \mathbb{Z}_q^n$, $\bar{B} = q^{-1}B$ implies that $\|\bar{B}^*\| = \|B^*\|/q$ where the factor q is exactly $\#G^{1/n}$: this suggests that in general, we might hope to reduce $\|\bar{B}^*\|$ by a factor close to $\#G^{1/n}$, compared to $\|B^*\|$.

Another trivial case of structural lattice reduction is $G = \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_n}$ where the q_i's are distinct positive integers of similar bit-length. If $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is a basis of $L \subseteq \mathbb{Z}^n$, then $\bar{B} = (q_1^{-1}\mathbf{b}_1, \ldots, q_n^{-1}\mathbf{b}_n)$ generates an overlattice $\bar{L}$ such that $\bar{B}^* = (q_1^{-1}\mathbf{b}_1^*, \ldots, q_n^{-1}\mathbf{b}_n^*)$, and therefore $\|\bar{B}^*\| \leq \|B^*\|/\min_{i=1}^n q_i$. The factor $\min_{i=1}^n q_i$ is close to $\#G^{1/n}$ if the q_i's have similar bit-length. But if the q_i's are unbalanced, such as when $\min_{i=1}^n q_i = 1$, then the bound is much weaker. In particular, the case $G = \mathbb{Z}_p$ for some large prime p looks challenging, as the trivial choice $\bar{B} = (p^{-1}\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ looks useless: $\bar{L}/L \simeq G$ but $\|\bar{B}^*\|$ is likely to be essentially as big as $\|B^*\|$, because for a typical reduced basis, the first $\|\mathbf{b}_i^*\|$'s have the same size.

4.2 Co-cyclic Lattice Reduction

As a warm-up, we solve structural lattice reduction when the target group G is cyclic of order q, which we call *co-cyclic lattice reduction.* Let $\bar{B}$ be a solution of structural reduction on $(L(B), G, \sigma)$: $C = (q\bar{b}_1, \bar{b}_2, \ldots, \bar{b}_n)$ is a basis of L s.t. $\|\mathbf{c}_1\| \leq q\sigma$ and $\|\mathbf{c}_i^*\| \leq \sigma$ for all $i \geq 2$.

To find such a basis $\bar{B}$, we first show how to transform B to ensure $\|\mathbf{b}_i^*\| \leq \sigma$ for all $i \geq 2$, using a poly-time algorithm which we call *unbalanced reduction* (see Alg. 2). This algorithm can be explained as follows: in dimension two, it is easy to make $\mathbf{b}_2^*$ arbitrarily short by lengthening $\mathbf{b}_1$ (adding a suitable multiple of $\mathbf{b}_2$), since $\|\mathbf{b}_1\| \times \|\mathbf{b}_2^*\| = \mathrm{vol}(L)$ is invariant. Unbalanced reduction works by iterating this process on two-dimensional projected lattices, similarly to the classical size-reduction process. However, one would like to make sure that the resulting first basis vector $\mathbf{c}_1$ does not become too large, which is quantified by the following result:

Theorem 4.1 (Unbalanced reduction) *Given an n-dim projected block $B = B'_{[i,i+n-1]}$ of a lattice $L \subseteq \mathbb{Z}^m$ and a target $\sigma \in \mathbb{Q}^+$, Alg. 2 outputs in polynomial time an $n \times n$ unimodular matrix U such that $C = UB$ satisfies $\|\mathbf{c}_1\| \leq n\sigma\delta_\sigma(B)$ and $\|\mathbf{c}_i^*\| \leq \sigma$ for $i \geq 2$, and:*

$$\delta_\nu(B) \;\leq\; \delta_\nu(C) \;\leq\; \frac{\|\mathbf{c}_1\|}{\sigma\delta_\sigma(B)} \times \delta_\nu(B) \;\textit{ for all } \nu \leq \sigma \tag{1}$$

$$\textit{where } \delta_\sigma(B) \underset{def}{=} \prod_{j=1}^{n} \max\left(1,\ \left\|\mathbf{b}_j^*\right\|/\sigma\right). \tag{2}$$

We call $\delta_\sigma(B)$ the *cubicity-defect* of B relatively to σ: it basically measures by which amount the hypercube of side σ should be scaled up to cover the parallelepiped spanned by $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$. The proofs of Th. 4.1 and Alg. 2 can be found in App. C.2. Th. 4.1 shows that Alg. 2 solves co-cyclic lattice reduction for $q \geq n\delta_\sigma(B)$. However, this may not be suitable for our applications, since this lower bound depends on B and might be unbounded. To address this issue, we now show that LLL can bound $\delta_\sigma(B)$ depending only on n for appropriate σ:

Algorithm 2 Unbalanced Reduction

```
Input: an $n \times m$ basis $B$ of an integer lattice $L \subseteq \mathbb{Z}^m$ and a target length $\sigma \in \mathbb{Q}^+$. More generally, $B$ can be any
    $n$-dimensional projected block $B = B'_{[i,i+n-1]}$ of some basis $B'$ of $L \subseteq \mathbb{Z}^m$.
Output: an $n \times n$ unimodular matrix $U$ such that $C = UB$ satisfies $\|c_i^*\| \leq \sigma$ for $i \geq 2$ and $\|\mathbf{c}_1\| \leq n\sigma\delta_\sigma(B)$.
$C \leftarrow B$, $U \leftarrow I_n$ and compute the Gram-Schmidt matrices $\mu$ and $C^*$
If $\|\mathbf{c}_i^*\| \leq \sigma$ for all $i$, return $U$
for $i = k-1$ downto 1 where $k$ is the largest index such that $\|\mathbf{c}_k^*\| > \sigma$ do
    if $\|\mathbf{c}_i^*\| \leq \sigma$ then
        $\alpha \leftarrow \lfloor -\mu_{i+1,i} \rceil$
    else
        $\alpha \leftarrow \left\lceil -\mu_{i+1,i} + \frac{\|\mathbf{c}_{i+1}^*\|}{\|\mathbf{c}_i^*\|}\sqrt{(\|\mathbf{c}_i^*\|/\sigma)^2 - 1} \right\rceil$
    end if
    $(\mathbf{c}_i, \mathbf{c}_{i+1}) \leftarrow (\mathbf{c}_{i+1} + \alpha \cdot \mathbf{c}_i,\ \mathbf{c}_i)$, $(\mathbf{u}_i, \mathbf{u}_{i+1}) \leftarrow (\mathbf{u}_{i+1} + \alpha \cdot \mathbf{u}_i,\ \mathbf{u}_i)$ and update the GS matrices $\mu$ and $C^*$.
end for
return $U$
```

Theorem 4.2 (LLL's cubicity-defect) *Let L be a full-rank lattice in $\mathbb{R}^n$ and $\sigma \geq ((1 + \varepsilon_{LLL})\sqrt{4/3})^r \cdot \mathrm{bl}(L)$ for some $r \geq 0$. If B is an LLL-reduced basis of L with factor ε_{LLL}, then $\delta_\sigma(B) \leq ((1 + \varepsilon_{LLL})\sqrt{4/3})^{\frac{(n-2r)^2}{8} + \frac{(n-2r)}{4}}$.*

By combining Th. 4.1 and 4.2, we obtain:

Theorem 4.3 (Co-cyclic Reduction) *Given an $n \times m$ basis of a lattice $L \subseteq \mathbb{Z}^m$, $\varepsilon > 0$ and a rational $\sigma \geq ((1+\varepsilon_{LLL})\sqrt{4/3})^r \cdot \mathrm{bl}(L)$ for some $r \geq 0$, and an integer $q \geq n((1+\varepsilon_{LLL})\sqrt{4/3})^{\frac{(n-2r)^2}{8}+\frac{(n-2r)}{4}}$, Alg. 3 computes a basis $\bar{B}$ of an overlattice $\bar{L} \supseteq L$ in time polynomial in the basis size, σ and $1/\varepsilon$, such that $\left\|\bar{B}^*\right\| \leq \sigma$ and $(q\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_n)$ is a basis of L. In particular, $\bar{L}/L \simeq \mathbb{Z}_q$.*

For instance, Th. 4.3 with $r = n$ implies that given a lattice L and any cyclic group G of sufficiently large order (*i.e.* $2^{\Omega(n^2)}$), one can efficiently obtain a basis $\bar{B}$ of some overlattice $\bar{L}$ of L such that $\bar{L}/L \simeq G$ and $\|\bar{B}^*\| \leq \mathrm{bl}(L)$: by comparison, an LLL-reduced basis only approximates $\mathrm{bl}(L)$ to some exponential factor in the worst case.

Algorithm 3 Co-cyclic Reduction

```
Input: a basis of a full-rank integer lattice $L \subseteq \mathbb{Z}^n$, a factor $\varepsilon > 0$, and a rational $\sigma \geq ((1+\varepsilon_{\mathrm{LLL}})\sqrt{4/3})^r \cdot \mathrm{bl}(L)$ for
    some $r \geq 0$, and an integer $q \geq n((1+\varepsilon_{\mathrm{LLL}})\sqrt{4/3})^{\frac{(n-2r)^2}{8}+\frac{(n-2r)}{4}}$
Output: a basis $\bar{B}$ of an overlattice $\bar{L}$ such that $\|\bar{B}^*\| \leq \sigma$ and $\bar{L}/L \simeq \mathbb{Z}_q$.
Apply Alg. 2 on an LLL-reduced basis with factor $\varepsilon_{\mathrm{LLL}}$ output by the LLL algorithm.
return $\bar{B} = (\frac{\mathbf{c}_1}{q}, \mathbf{c}_2, \ldots, \mathbf{c}_n)$ where $C$ is the basis of $L$ returned by Alg. 2.
```

4.3 Arbitrary Groups

Using unbalanced reduction, we prove that for an arbitrary sufficiently large finite abelian group G of rank $\leq n$, given any basis B of the lattice $L \subseteq \mathbb{Z}^n$, one can compute a basis $\bar{B}$ of some overlattice $\bar{L}$ of L s.t. $\bar{L}/L \simeq G$ effectively and $\|\bar{B}^*\|$ is essentially lower than $\|B^*\|/\#G^{1/n}$. In particular, $\mathrm{bl}(\bar{L})$ is essentially $\#G^{1/n}$ smaller than $\mathrm{bl}(L)$. Although this is slightly weaker than the result we obtained (in the previous subsection) for cyclic groups G, it is sufficient for our worst-case to average-case reductions.

Theorem 4.4 (Structural Lattice Reduction) *Given an $n \times m$ basis B of a lattice $L \subseteq \mathbb{Z}^n$, and $k \leq n$ integers $q_1 \geq \cdots \geq q_k$ defining the group $G = \prod_{i=1}^k \mathbb{Z}_{q_i}$ s.t. $n^k(\|B^*\|/\sigma)^n \leq \#G$ or:*
$\#G \geq \frac{n!}{(n-k)!}\delta_\sigma(B)$ and for all $i \leq k$, $\|B^*\|/\sigma \leq q_i/(n+1-i)$
Alg. 4 outputs a basis $\bar{B}$ of an overlattice $\bar{L} \supseteq L$ such that $\left\|\bar{B}^*\right\| \leq \sigma$ and $(q_1\bar{\mathbf{b}}_1, \ldots, q_n\bar{\mathbf{b}}_n)$ is a basis of L where $q_i = 1$ for $i > k$. In particular, $\bar{L}/L \simeq G$.

For instance, the condition $n^k(\|B^*\|/\sigma)^n \leq \#G$ in Th. 4.4 means that σ (and therefore $\|\bar{B}^*\|$) can be chosen as low as $n^{k/n}\|B^*\|/(\#G)^{1/n}$. The proof of Th. 4.4 can be found in App. C.3.

Algorithm 4 Structural Lattice Reduction

```
Input: $\sigma$, an $n \times m$ basis $B$ of an integer lattice $L$, and $(q_1, \ldots, q_k)$ s.t. $G = \prod_{i=1}^k \mathbb{Z}_{q_i}$ satisfies the conditions of Th. 4.4
Output: an $n \times m$ basis $\bar{B}$ of an overlattice $\bar{L}$ of $L$ such that $\|\bar{B}^*\| \leq \sigma$ and $\bar{L}/L \simeq G$.
$C \leftarrow B$
for $i = 1$ to $k$ do
    if $\left\|C^*_{[i,n]}\right\| \leq \sigma$ return $\bar{B} = (\frac{\mathbf{c}_1}{q_1}, \ldots, \frac{\mathbf{c}_k}{q_k}, \mathbf{c}_{k+1}, \ldots, \mathbf{c}_n)$
    Compute the smallest $\ell \geq \sigma$ such that $\ell \cdot \delta_\ell(C_{[i,n]}) = q_i\sigma/(n-i+1)$.
    $V \leftarrow$ UnbalancedReduction($C_{[i,n]}, \sigma$) using Alg. 2.
    Apply $V$ on $(\mathbf{c}_i, \ldots, \mathbf{c}_n)$
end for
return $\bar{B} = (\frac{\mathbf{c}_1}{q_1}, \ldots, \frac{\mathbf{c}_k}{q_k}, \mathbf{c}_{k+1}, \ldots, \mathbf{c}_n)$
```

Intuitively, Alg. 4 simply applies unbalanced reduction iteratively, cycle by cycle of G.

4.4 Application

Structural reduction finds a short overlattice basis, which can typically be used to sample short (overlattice) vectors, and which provides the following effective isomorphisms:

Proposition 4.5 *Let L and $\bar{L}$ be two full-rank lattices such that $\bar{L} \supseteq L$ and $\bar{L}/L \simeq G$ where $G = \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_k}$. Given bases B and $\bar{B}$ of resp. L and $\bar{L}$, one can compute in polynomial time a morphism φ s.t. the sequence $0 \to L \xrightarrow{\mathrm{id}} \bar{L} \xrightarrow{\varphi} G \to 0$ is exact, and a "dual" morphism $\varphi^\times : L^\times \to \hat{G}$ s.t.*

$$[\varphi^\times(\mathbf{u})](\varphi(\mathbf{v})) = \langle \mathbf{u}, \mathbf{v} \rangle \mod 1 \textit{ for all } \mathbf{u} \in L^\times \textit{ and all } \mathbf{v} \in \bar{L} \tag{3}$$

Furthermore, preimages of $\varphi^\times$ can be computed in polynomial time.

Proof. (Sketch) Let $(e_1, \ldots, e_k)$ be the canonical generators of $G = \prod_{i=1}^k \mathbb{Z}_{q_i}$. Find any basis C of L and $\bar{C}$ of $\bar{L}$ such that $C = (q_1\bar{\mathbf{c}}_1, \ldots, q_k\bar{\mathbf{c}}_k, \bar{\mathbf{c}}_{k+1}, \ldots \bar{\mathbf{c}}_n)$, then let φ be the morphism mapping C to $(e_1, \ldots, e_k, 0, \ldots, 0)$ and $\varphi^\times$ be the mapping from $C^\times$ to $(\hat{e}_1, \ldots, \hat{e}_k, \hat{0}, \ldots, \hat{0})$. □

This proposition still holds if G is an explicit finite abelian group.

5 Hardness of Group-SIS

Our hardness result for GSIS requires that the finite abelian group G is *explicit* (see Sect. 2).

5.1 Overview

We first sketch how to adapt the SIS reduction to GSIS using structural lattice reduction.

The main idea behind the SIS reduction can be traced back to 1935, when Mordell [29] published an arithmetical proof of Minkowski's theorem. To prove the existence of short vectors in a full-rank lattice $L \subseteq \mathbb{R}^n$, Mordell implicitly presented an algorithm to find short vectors from (exponentially many) long vectors, as follows. Let $q \geq 1$ be an integer and $\mathbf{w}_1, \ldots, \mathbf{w}_m \in L$ be distinct vectors of norm $\leq R$, where $m > q^n$: for large R, m can be essentially chosen as large as the volume of the R-radius ball divided by the volume of L. Let $\mathbf{v}_i = q^{-1}\mathbf{w}_i \in q^{-1}L$. Since $m > q^n = [(q^{-1}L) : L]$, there are $i \neq j$ such that $\mathbf{v}_i \equiv \mathbf{v}_j \bmod L$, *i.e.* $\mathbf{v}_i - \mathbf{v}_j = q^{-1}(\mathbf{w}_i - \mathbf{w}_j) \in L$ whose (nonzero) norm is $\leq 2R/q$, which is short for appropriate choices of q and R.

This algorithm is not efficient since m is exponential in q, but it can be made polynomial by reducing m to poly(n), using a SIS(m, n, q) oracle. Indeed, let L be a full-rank integer lattice in $\mathbb{Z}^n$. The lattice $\bar{L} = q^{-1}L$ is an overgroup of L such that $\bar{L}/L \simeq \mathbb{Z}_q^n = G$: namely, there is an exact sequence of groups $0 \to L \xrightarrow{\mathrm{id}} \bar{L} \xrightarrow{\varphi} G \to 0$, where φ is efficiently computable, *e.g.* for any fixed basis $(\bar{\mathbf{b}}_1, \ldots, \bar{\mathbf{b}}_n)$ of $\bar{L}$, let $\varphi(\sum_{i=1}^n x_i\bar{\mathbf{b}}_i) = (x_1 \bmod q, \ldots, x_n \bmod q) \in G$.

Furthermore, if $\bar{B}$ is short enough compared to the minima of L, it is possible to sample short vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \bar{L}$ with Gaussian distribution of parameter as small as $\eta_\varepsilon(L)$. Fourier analysis guarantees that for such Gaussian distributions, each projection $g_i = \varphi(\mathbf{v}_i)$ is uniformly distributed over G. This allows us to call an SIS oracle on $(g_1, \ldots, g_m)$, which outputs a short $\mathbf{x} \in \mathbb{Z}^m$ such that $\sum_{i=1}^m x_i g_i = 0$, *i.e.* $\sum_{i=1}^m x_i\varphi(\mathbf{v}_i) = 0$ which implies that $\mathbf{v} = \sum_{i=1}^m x_i\mathbf{v}_i \in L$. This $\mathbf{v}$ can be proved to be non-zero with overwhelming probability, and it is short because the $\mathbf{v}_i$'s and $\mathbf{x}$ are short, which concludes the reduction from worst-case SIVP to SIS.

With this formalization, we can replace the SIS oracle by a GSIS oracle, as while as we are able to sample short vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \bar{L}$ with Gaussian distribution, where $\bar{L}/L \simeq G$. And this is exactly what structural lattice reduction ensures. Previous SIS reductions used special choices of $\bar{L}$ and a different way to sample short vectors in the overlattice: see App. H for more information.

5.2 Reducing Worst-case ApproxSIVP to GSIS

Our main result formalizes the previous sketch and states that for appropriate choices of (G, m, β), if one can solve $\mathrm{GSIS}(G, m, \beta)$ on average, then one can approximate SIVP in the worst case, *i.e.* one can efficiently find short vectors in every n-dimensional lattice:

Theorem 5.1 *Let $n \in \mathbb{N}$ and $\varepsilon = \mathrm{negl}(n)$. Given as input a basis B of a full-rank integer lattice $L \subseteq \mathbb{Z}^n$ and $\sigma \geq \sqrt{2}\,\mathrm{bl}(L)$, and an explicit finite abelian group G of rank $k \leq n$ such that $\#G \geq n^k(\|B^*\|/\sigma)^n$, Alg. 5 outputs (in random polynomial time) n linearly independent vectors of L with norm $\leq \sigma\eta_\varepsilon(\mathbb{Z}^n)\sqrt{n\pi}\beta$, using polynomially many calls to an oracle solving $\mathrm{GSIS}(G, m, \beta)$ with probability $\geq 1/\mathrm{poly}(n)$.*

Algorithm 5 Reducing ApproxSIVP to GSIS

```
Input: a basis $B$ of a full-rank integer lattice $L \in \mathbb{Z}^n$, a parameter $\sigma \geq \sqrt{2}\,\mathrm{bl}(L)$, a negl. $\varepsilon > 0$, an explicit finite abelian
    group $G$ satisfying the condition of Th. 5.1, and an oracle $\mathcal{O}$ solving $\mathrm{GSIS}(G, m, \beta)$ with probability $\geq 1/\mathrm{poly}(n)$.
Output: A set $S$ of $n$ linearly independent vectors of $L$ of norm $\leq \sigma\eta_\varepsilon(\mathbb{Z}^n)\sqrt{n/2\pi}\beta$.
$S \leftarrow \emptyset$.
Call structural reduction (Alg. 4) on $(B, G, \sigma)$ to get $\bar{B}$ s.t. $\|\bar{B}^*\| \leq \sigma$ and $\varphi : \bar{L} \to G$ (Prop. 4.5) where $\bar{L} = L(\bar{B})$.
repeat
    Sample $\mathbf{v}_1, \cdots, \mathbf{v}_m \in \bar{L}$ with distribution $D_{\bar{L},\sigma\eta_\varepsilon(\mathbb{Z}^n),\mathbf{0}}$ using $\bar{B}$.
    $g_i = \varphi(\mathbf{v}_i)$ for $1 \leq i \leq m$, forming a sequence $\mathbf{g} = (g_1, \ldots, g_m) \in G^m$.
    Call the GSIS-oracle $\mathcal{O}$ on $\mathbf{g}$, which returns $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{Z}^m$ s.t. $\sum_{i=1}^m x_i g_i = 0$.
    $\mathbf{v} \leftarrow \sum_{i=1}^m x_i \mathbf{v}_i \in L$
    if $\|\mathbf{v}\| \leq \sigma\eta_\varepsilon(\mathbb{Z}^n)\sqrt{n\pi}\beta$ and $\mathbf{v} \notin \mathrm{span}(S)$ then $S \leftarrow S \cup \{\mathbf{v}\}$
until $\dim(S) = n$
Return $S$
```

In particular, by letting $\sigma = \frac{\|B\|^*}{2\eta_\varepsilon(\mathbb{Z}^n)\sqrt{n/\pi\beta}}$, we can obtain an incremental version of the reduction, where the output basis is twice as short as the input. This generalizes [28, Th. 5.9] and [20, Th. 9.2] with a GSIS oracle instead of SIS. Iterating Th. 5.1 until we reach $\sigma = \sqrt{2}\,\mathrm{bl}(L)$ allows to connect the average-case hardness of GSIS to the worst-case of ApproxSIVP.

Corollary 5.2 *Let $n \in \mathbb{N}$ and $\varepsilon = \mathrm{negl}(n)$. Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of explicit finite abelian groups s.t. $\#G_n \leq (\beta_n/\sqrt{m_n})^{m_n}$ for $m_n \in \mathbb{N}$ and G_n has rank k_n. If $\#G_n \geq n^{k_n}\left(\eta_\varepsilon(\mathbb{Z}^n)\sqrt{2n/\pi}\beta_n\right)^{\max(n,k_n)}$, then using polynomially many calls to an oracle solving $\mathrm{GSIS}(G_n, m_n, \beta_n)$ with probability $\geq 1/\mathrm{poly}(n)$, one can solve worst-case n-dimensional $\mathrm{ApproxSIVP}_{\eta_\varepsilon(\mathbb{Z}^n)\sqrt{n/\pi}\beta_n}$ in (randomized) polynomial time.*

Consider the set of all full-rank integer lattices $\subseteq \mathbb{Z}^m$ of volume $\geq \omega_n = n^m\left(\eta_\varepsilon(\mathbb{Z}^n)\sqrt{2n/\pi}\beta_n\right)^m$. This set can be partitioned as $\cup_G \mathcal{L}_{G,m}$ where G runs over all finite abelian groups of order $\geq \omega_n$ and rank $\leq m$. Each such G satisfies the conditions of Cor. 5.2, and therefore GSIS over G is as hard as worst-case lattice problems: for any of the partition cells $\mathcal{L}_{G,m}$, finding short vectors in a random lattice from this cell is as hard as finding short vectors in any n-dim lattice.

6 Hardness of Decisional-Group-LWE

In this section, we show how to transfer the following Decisional-LWE hardness results to Decisional-GLWE:

Theorem 6.1 ([38, 34]) *Let $n \in \mathbb{N}$ and $q_n \geq 1$ be a sequence of integers, and let $\alpha_n \in (0, 1)$ be a real sequence such that $\alpha_n q_n \geq 2\sqrt{n}$. There exists a quantum reduction from worst-case*

n-dimensional $\mathrm{GapSVP}_{\tilde{O}(n/\alpha_n)}$ *to Decisional-*$\mathrm{GLWE}_{\mathbb{Z}_{q_n}^n,\alpha_n}$*. If $q_n \geq 2^{n/2}$ is smooth then there is a classical reduction between them.*

Theorem 6.2 ([12]) *Let $n \in \mathbb{N}$ and $q_n \geq 1$ be a sequence of integers, and let $\alpha_n \in (0,1)$ be a real sequence such that $\alpha_n \geq 2n^{1/4}/2^{\sqrt{n}/2}$. There exists a classical reduction from worst-case $\sqrt{n}$-dimensional* $\mathrm{GapSVP}_{\tilde{O}(\sqrt{n}/\alpha_n)}$ *to Decisional-*$\mathrm{GLWE}_{\mathbb{Z}_{q_n}^n,\beta_n}$*, where $\beta_n^2 = 10n\alpha_n^2 + \frac{n}{q_n^2} \cdot \omega(\log n)$*

To do so, we reduce Decisional-LWE to Decisional-GLWE using a technique we call group switching. This technique transforms GLWE samples over a group G to another group G', generalizing the modulus-dimension switching technique in [12], which is the special case $G = \mathbb{Z}_q^n$ and $G' = \mathbb{Z}_{q'}^{n'}$. We believe that the group switching technique proposed below is useful to better understand the core idea of the modulus-dimension switching technique.

Before presenting group switching, we note that the modulus-dimension switching technique from [12] implicitly uses a special case of structural lattice reduction. More precisely, Brakerski *et al.* [12] defined a special lattice Λ (see Th 3.1 of [12]) to transform LWE samples over $G = \mathbb{Z}_q^n$ to LWE samples over $G' = \mathbb{Z}_{q'}^{n'}$, but the meaning of Λ may look a bit mysterious. The lattice Λ is defined as $\Lambda = \frac{1}{q'}\mathbb{Z}^{n'} \cdot H + \mathbb{Z}^n$ where H is some $n' \times n$ integer matrix: this matrix is actually denoted by G in [12], but this would collide with our notation G for finite abelian groups. And [12] provided a good basis of Λ in special cases. We note that the exact definition of Λ is not important: the quotient $\Lambda/\mathbb{Z}^n$ turns out to be isomorphic to the group $G' = \mathbb{Z}_{q'}^{n'}$, as shown by the transformation mapping $\frac{1}{q'}\mathbf{x} \cdot H + \mathbf{y} \in \Lambda$ to $\mathbf{x} \mod q' \in G'$. Thus, finding a good basis of Λ is actually a special case of structural lattice reduction for the lattice $\mathbb{Z}^n$ and the group G'. Therefore, it is natural to use structural lattice reduction directly (instead of an ad-hoc process) to obtain a more general statement than the modulus-dimension switching technique of [12].

Since we have two groups G and G' and two overlattices $\bar{L}$ and $\bar{L}'$ of $\mathbb{Z}^n$, we will have two morphisms $\varphi : \bar{L} \to G$ and $\varphi' : \bar{L}' \to G'$ with $\ker(\varphi) = \ker(\varphi') = \mathbb{Z}^n$. Both morphisms are associated to their dual morphism as in Prop. 4.5, *i.e.* $\varphi^\times : \mathbb{Z}^n \to \hat{G}$ and $\varphi'^\times : \mathbb{Z}^n \to \hat{G}'$, satisfying $[\varphi'^\times(\mathbf{u})](\varphi'(\mathbf{v})) = \langle \mathbf{u}, \mathbf{v} \rangle \mod 1$ for all $\mathbf{u} \in \mathbb{Z}^n$ and all $\mathbf{v} \in \bar{L}'$ (resp. without primes). These morphisms are summarized in Figure 1.

We say that a distribution S over $\mathbb{Z}^n$ is K-bounded if $\Pr_{\mathbf{s}\leftarrow S}[\|\mathbf{s}\| > K] \leq \mathrm{negl}(n)$. By extension, given a (public) morphism f from $\mathbb{Z}^n$ to $\hat{G}$, we say that a distribution $\mathcal{S}$ over $\hat{G}$ is K-bounded (for f) if it is the image of a K-bounded distribution by f. This in particular means, that for almost all choices of f[1], every distribution on G including the uniform one are by definition automatically $\#G^{1/n}$-bounded. In the following, we will choose $\varphi^\times = f$ and φ its dual morphism accordingly. Thus, any secret $\hat{s} \leftarrow \mathcal{S}$ has with overwhelming probability a preimage $\mathbf{s} \in \mathbb{Z}^n$ of norm $\leq K$. Note that the small $\mathbf{s} \in \mathbb{Z}^n$ may be hard to compute from $\hat{s}$, however in this case, what matters is its existence. During group switching, the new secret in $\hat{G}'$ will be $\varphi'^\times(\mathbf{s})$, and the new K-bounded distribution $\mathcal{S}' = \varphi'^\times(S)$.

Lemma 6.3 (Group Switching) *Let G and G' be two finite abelian groups of rank $\leq n$ s.t. G is fully-explicit and G' is explicit. Let $\bar{L}$ be an overlattice of $\mathbb{Z}^n$ such that $\bar{L}/\mathbb{Z}^n \simeq G$. Let $\bar{B}'$ be a basis of an overlattice $\bar{L}'$ of $\mathbb{Z}^n$ such that $\bar{L}'/\mathbb{Z}^n \simeq G'$. Let φ, φ' and $\varphi'^\times$ be defined as in Prop. 4.5. Let $r \geq \max\left(\sqrt{2}\eta_\varepsilon(\bar{L}), \|\bar{B}'^*\| \cdot \eta_\varepsilon(\mathbb{Z}^n)\right)$, where ε is some negligible function. Then, there is an efficient randomized algorithm which, given as input a sample from $G \times \mathbb{T}$, outputs a sample from $G' \times \mathbb{T}$, with the following properties:*

- If the input sample has uniform distribution in $G \times \mathbb{T}$, then the output sample has uniform distribution in $G' \times \mathbb{T}$ (except with negligible distance).

[1] Ideally, f should be collision resistant among samples from S. In the classical LWE ($G = \mathbb{Z}_q^n$), f would map $\mathbf{s} \in \mathbb{Z}^n$ to the secret character $\hat{s} : \mathbf{y} \to {}^1/_q\langle \mathbf{s}, \mathbf{y} \rangle \mod 1$ in $\hat{G}$.

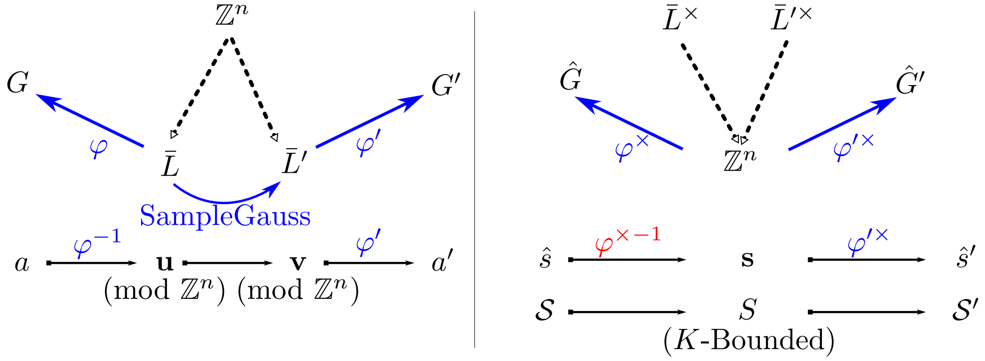


Figure 1: Main components of group switching and their relationship

Dotted arrows represent overlattices. The left side are the primal extensions, the right side contains dual overlattices. On the left of each diagram, we recognize the dual pair of surjective morphisms φ from $\bar{L}$ (mod $\mathbb{Z}^n$) to G, and $\varphi^\times$ from $\mathbb{Z}^n$ (mod $\bar{L}^\times$) to $\hat{G}$. On the right side of each diagram, the dual morphisms $\varphi', \varphi'^\times$ related to the second group G'.
To transform a $A_{G,\alpha}(\hat{s})$ sample (a, b) to a $A_{G',\beta}(\hat{s}')$ sample (a', b'), we compute $\mathbf{u} \in \varphi^{-1}(a)$ a preimage in $\bar{L}\,(\text{mod}\,\mathbb{Z})^n$, we sample a point $\mathbf{v}$ in $\bar{L}'\,(\text{mod}\,\mathbb{Z})^n$ close to $\mathbf{u}$, and we apply φ' to get a'.
On the dual side, the secret $\hat{s}$ admits in general a unique preimage $\mathbf{s}$ by $\varphi^\times$ which has small norm. Finding this small $\mathbf{s}$ may in some cases be very difficult, as it requires to solve a CVP in $\bar{L}^\times$. However, only the existence of the small $\mathbf{s}$ matters. Then, the new (unknown) secret is $\hat{s}' = \varphi'^\times(\mathbf{s})$. In general, we know the K-bounded distribution S over $\mathbb{Z}^n$ from which $\mathcal{S} = \varphi^\times(S)$ is defined. Thus, the distribution of secrets $\mathcal{S}' = \varphi'^\times(S)$ in the new GLWE is entirely determined.
Finally, note that by duality of the morphisms, $\hat{s}(a) = \langle \mathbf{s}, \mathbf{u} \rangle \approx \langle \mathbf{s}, \mathbf{v} \rangle = \hat{s}'(a')$. We just need to add some noise $b' = b$+noise to compensate the Gaussian approximation between $\langle \mathbf{s}, \mathbf{u} \rangle$ and $\langle \mathbf{s}, \mathbf{v} \rangle$ in the middle.

- *If the input is distributed according to $A_{G,\alpha}(\hat{s})$ for some $\hat{s} = \varphi^{\times}(\mathbf{s})$ s.t. $\mathbf{s} \in \mathbb{Z}^n$ and $\|\mathbf{s}\| \leq K$, then the output distribution is statistically close to $A_{G',\beta}(\hat{s}')$, where $\hat{s}' = \varphi'^{\times}(\mathbf{s}) \in \hat{G}'$ and $\beta^2 = \alpha^2 + r^2(\|\mathbf{s}\|^2 + K^2) \leq \alpha^2 + 2(rK)^2$.*

By combining Group Switching (Lemma 6.3) with structural reduction (Th. 4.4), one obtains the following reduction between Decisional-GLWE of two groups G and G':

Corollary 6.4 (GLWE to GLWE) *Let $n \in \mathbb{N}$ and $0 < \sigma_n < 1$ be a real sequence. Let $(G_n)_{n\in\mathbb{N}}$ and $(G'_n)_{n\in\mathbb{N}}$ be two sequences of finite abelian groups with respective rank $k_n \leq n$ and $k'_n \leq n$ s.t. $\#G_n \geq n^{k_n}(\sqrt{2}/\sigma_n)^n$ (or if $G_n = \mathbb{Z}_{q_n}^n$ where $q_n \geq \sqrt{2}/\sigma_n$) and $\#G'_n \geq n^{k'_n}(1/\sigma_n)^n$. Assume that G_n is fully-explicit and G'_n is explicit. Let S be an arbitrary K_n-bounded distribution over $\mathbb{Z}^n$ and $\mathcal{S} = \varphi^{\times}(S)$ its image by some morphism $\varphi^{\times} : \mathbb{Z}^n \to \hat{G}_n$, $\alpha_n, \beta_n > 0$ be two real sequences and $\varepsilon = \mathrm{negl}(n)$ satisfying $\beta_n^2 \geq \alpha_n^2 + 2(\sigma_n K_n \cdot \eta_\varepsilon(\mathbb{Z}^n))^2$. Then there is an efficient reduction from Decisional-*$\mathrm{GLWE}_{G_n,\leq\alpha_n}(\mathcal{S})$ *to Decisional-*$\mathrm{GLWE}_{G'_n,\leq\beta_n}(\mathcal{S}')$*, where $\mathcal{S}' = \varphi'^{\times}(S)$ for some morphism $\varphi'^{\times} : \mathbb{Z}^n \to \hat{G}'_n$*

Proof. Given the canonical basis of $\mathbb{Z}^n$ and the group G_n, structural reduction finds an overlattice $\bar{L}$ together with a basis $\bar{C}$ s.t. $\left\|\bar{C}^*\right\| \leq \sigma_n/\sqrt{2}$. Therefore $\sqrt{2}\eta_\varepsilon(\bar{L}) \leq \sigma_n \eta_\varepsilon(\mathbb{Z}^n)$. And structural reduction on G'_n and σ_n gives a short basis $\bar{B}'$ of length $\leq \sigma_n$ and defines $\bar{L}'$. The rest of the proof follows immediately from Lemma 6.3. □

Using the normal form [4] of LWE, namely, if $\mathcal{S}$ is the image of the $\alpha_n q_n \sqrt{n}$-bounded distribution $\mathcal{D}_{\mathbb{Z}^n,\alpha_n q_n}$, through the canonical embedding which maps $\mathbf{s} \in \mathbb{Z}^n$ to the character $\hat{s} = \mathbf{y} \to {}^1\!/_{q_n}\langle \mathbf{s}, \mathbf{y}\rangle \mod 1$, we obtain the quantum/classical hardness of Decisional-GLWE problem for any sufficiently large finite abelian group, together with Theorems 6.1 and 6.2:

Corollary 6.5 (Quantum Hardness of GLWE) *Let $n \in \mathbb{N}$ and $q_n \geq 1$ be a sequence of integers and $(G'_n)_{n\in\mathbb{N}}$ be a sequence of any finite abelian explicit groups such that $\#G'_n \geq n^{k_n}(q_n/\sqrt{2})^n$ where $k_n = \mathrm{rank}\,(G'_n) \leq n$. Let $\alpha_n, \beta_n \in (0,1)$ be two real sequences such that $\alpha_n q_n \geq 2\sqrt{n}$ and $\beta_n = \alpha_n \sqrt{n} \cdot \omega(\sqrt{\log n})$. Then there exists a quantum reduction from worst-case n-dimensional* $\mathrm{GapSVP}_{\tilde{O}(n/\alpha_n)}$ *to Decisional-*$\mathrm{GLWE}_{G'_n,\beta_n}$.

The lower bound on $\#G'_n$ is better than for $\#G_n$ in Cor. E.2 and 5.2, because group switching relies on structural reduction over $\mathbb{Z}^n$, rather than over an arbitrary lattice: the canonical basis of $\mathbb{Z}^n$ is orthonormal, which simplifies the bound of Sect. 4.

Corollary 6.6 (Classical Hardness of GLWE) *Let $n \in \mathbb{N}$ and $q_n \geq 1$ be a sequence of integers and $(G'_n)_{n\in\mathbb{N}}$ be a sequence of any finite abelian explicit groups such that $\#G'_n \geq n^{k_n}(q_n/\sqrt{2})^n$ where $k_n = \mathrm{rank}\,(G'_n) \leq n$. Let $\alpha_n, \beta_n \in (0,1)$ be two real sequences such that $\alpha_n \geq 2n^{1/4}/2^{\sqrt{n}/2}$ and $\beta_n^2 = n^2\alpha_n^2 \cdot \omega(\log n) + \frac{n^2}{q_n^2} \cdot \omega(\log^2 n)$. There exists a classical reduction from worst-case $\sqrt{n}$-dimensional* $\mathrm{GapSVP}_{\tilde{O}(\sqrt{n}/\alpha_n)}$ *to Decisional-*$\mathrm{GLWE}_{G'_n,\beta_n}$.

7 Abstracting Lattice Cryptography: Fully-Homomorphic Encryption from GLWE

We showed that GSIS and GLWE are hard under the same worst-case assumptions as SIS and LWE, provided that the group G is sufficiently large. This suggests to abstract lattice schemes based on SIS and/or LWE using an arbitrary finite abelian group G, and check that the security proof carries through under the assumption that GSIS or GLWE is hard. Such an abstraction may lead to a better understanding of the scheme and a clearer presentation: lattice schemes are typically described using matrices and vectors, which our abstraction avoids.

We illustrate this approach with fully-homomorphic encryption. First, we introduce a GLWE-based El Gamal-like encryption scheme, which generalizes Regev's LWE-based encryption [38] and its dual version [20]. Next, we extend this GLWE generalization of Regev's encryption into a somewhat-homomorphic encryption, by carefully abstracting the Alperin-Sheriff-Peikert variant [3] of the Gentry-Sahai-Waters homomorphic scheme [21]. In particular, we show how to evaluate any boolean function with a noise overhead proportional to the square root of its number of variables, how to recognize any regular language with a noise overhead proportional to the length of the tested word, and how to bootstrap the whole system with only a linear noise overhead instead of quadratic in [3].

7.1 A GLWE Variant of El Gamal Encryption

It is folklore that El Gamal encryption combines the one-time pad with Diffie-Hellman: the secret pad is a shared key between the person who encrypts and the holder of the secret key.

Diffie-Hellman uses a cyclic group G' (denoted additively), generated by some g' of order q. Let $f' : \mathbb{Z}_q \to G'$ be the "exponentiation" one-way function defined by $f(x') = x'.g'$: inverting f' is the DL problem over G'. The Diffie-Hellman map $\theta' : \mathbb{Z}_q \times \mathbb{Z}_q \to G'$ defined by $\theta'(a, b) = (ab).g'$ is bilinear. The pairing $\theta'(a, b)$ can be efficiently computed from (a, b), but also knowing only $(f'(a), b)$ or $(a, f'(b))$, which explains the DH key exchange: Alice picks $a \in_R \mathbb{Z}_q$ and discloses $f'(a)$, Bob picks $b \in_R \mathbb{Z}_q$ and discloses $f'(b)$, and both can compute the shared key $\theta'(a, b)$. Its security in the classical model covering passive adversaries is equivalent to DDH, which asks to distinguish $(f'(a), f'(b), \theta'(a, b))$ from $(f'(a), f'(b), c)$ where $a \in_R \mathbb{Z}_q$, $b \in_R \mathbb{Z}_q$ and $c \in_R G'$. In El Gamal encryption, the secret key is $x' \in_R \mathbb{Z}_q$, and the public key is $y' = f'(x') \in G'$. The ciphertext of a message $\mu' \in G'$ is $(c', d') \in G' \times G'$ where $c' = \mu' + z'.y'$, $d' = f'(z')$ where $z' \in_R \mathbb{Z}_q$ is a one-time key. The first entry c' is a one-time pad encryption of μ' with the DH key $z'.y' = \theta'(x', z')$. Breaking El Gamal's semantic security is equivalent to solving DDH.

By analogy, we first present a GLWE variant of Diffie-Hellman, with different one-way functions and pairing. We consider a (sufficiently large) finite abelian group G and $\mathbf{g} = (g_1, ..., g_m) \in G^m$ chosen uniformly at random. This defines two one-way functions:

- Let $f_{\mathbf{g}} : \mathbb{Z}^m \to G$ be the morphism defined by $f_{\mathbf{g}}(\mathbf{x}) = \sum_{i=1}^m x_i.g_i$, where $x_i.g_i$ is defined by the $\mathbb{Z}$-module structure of G. For suitable input distributions $\mathcal{D}$, such as the uniform distribution over $\{0, 1\}^m$ or some well-chosen discrete Gaussian distribution, the distribution of $f_{\mathbf{g}}(\mathbf{x})$ becomes statistically close to uniform (*e.g.* see the left-over-hash lemma), and $f_{\mathbf{g}}$ becomes one-way under GSIS.
- Let $f^\times_{\mathbf{g}} : \hat{G} \times \mathbb{T}^m \to \mathbb{T}^m$ defined by $f^\times_{\mathbf{g}}(\hat{s}, \mathbf{e}) = (\hat{s}(g_1)+e_1, \dots, \hat{s}(g_m)+e_m)$: if $\hat{s} \in_R \hat{G}$ and $\mathbf{e}$ is sampled from a suitable distribution such as $\mathcal{D}^m_\alpha$, then inverting $f^\times_{\mathbf{g}}(\hat{s}, \mathbf{e})$ is search-GLWE, and distinguishing $f^\times_{\mathbf{g}}(\hat{s}, \mathbf{e})$ from random is decisional-GLWE.

Now, consider the bilinear map $\theta : \hat{G} \times \mathbb{Z}^m \to \mathbb{T}$ defined by $\theta(\hat{s}, \mathbf{x}) = \hat{s}(f_{\mathbf{g}}(\mathbf{x}))$. Again, $\theta(\hat{s}, \mathbf{x})$ can be efficiently computed from $(\hat{s}, \mathbf{x})$. But it can also be computed knowing only $(\hat{s}, f_{\mathbf{g}}(\mathbf{x}))$ by definition, and it can be computed approximately knowing only $(f^\times_{\mathbf{g}}(\hat{s}, \mathbf{e}), \mathbf{x})$ by $\sum_{i=1}^m c_i x_i$ (where $\mathbf{c} = f^\times_{\mathbf{g}}(\hat{s}, \mathbf{e})$), provided that $\mathbf{e}$ and $\mathbf{x}$ are sampled from suitable distributions.

This motivates the GLWE noisy key exchange where Alice and Bob each compute their own approximation of $\theta(\hat{s}, \mathbf{x})$: Alice picks $\mathbf{x} \in \mathbb{Z}^m$ from some suitable distribution $\mathcal{D}$, and discloses $y = f_{\mathbf{g}}(\mathbf{x})$; Bob picks $\hat{s} \in_R \hat{G}$ and $\mathbf{e}$ from the distribution $\mathcal{D}^m_\alpha$, and discloses $\mathbf{c} = f^\times_{\mathbf{g}}(\hat{s}, \mathbf{e})$. Alice computes her key as $\sum_{i=1}^m c_i x_i$, and Bob computes his key as $\hat{s}(y) + e$ where e is sampled from $\mathcal{D}_\alpha$. Both keys are close to $\theta(\hat{s}, \mathbf{x})$. But, as opposed to Diffie-Hellman, Alice and Bob do not have symmetric roles, which leads to two El Gamal cryptosystems by swapping Alice and Bob: this is why Regev encryption has a so-called dual variant [20]. We now give a detailed description of the main cryptosystem, which generalizes Regev's [38], and which we use in our fully-homomorphic encryption.

Define the group $H = G \times \mathbb{T}_k$ where $k \in \mathbb{N}^+$ and $\mathbb{T}_k = \frac{1}{2^k}\mathbb{Z}/\mathbb{Z} \subseteq \mathbb{T}$ is a discretized torus.

$\texttt{GLWE.Gen}(1^n)$: Takes as input a security parameter n, it chooses a Gaussian parameter $0 < \alpha < 1$, a (sufficiently large) finite abelian group G and $m \in \mathbb{N}$. Choose $\mathbf{g} = (g_1, ..., g_m) \in_R G^m$, $\hat{s} \in_R \hat{G}$ and m Gaussian samples $e_1, ..., e_m \leftarrow \mathcal{D}_\alpha$. Set the public key $pk = (\mathbf{g}, \mathbf{y}) \in G^m \times \mathbb{T}_k^m$, where $y_i = \hat{s}(g_i) + e_i \in \mathbb{T}$, and the secret key $sk = \hat{s}$, *i.e.* $\mathbf{y} = f_{\mathbf{g}}^{\times}(\hat{s}, \mathbf{e})$.

$\texttt{GLWE.Enc}(pk, \mu)$: Takes as input the public key $pk = (\mathbf{g}, \mathbf{y}) \in G^m \times \mathbb{T}_k^m$ and a message $\mu \in \{0, 1\}$. It selects $\mathbf{x} = (x_1, ..., x_m) \in_R \{0, 1\}^m$, and returns $(d, c) \in H$, where $d = f_{\mathbf{g}}(\mathbf{x}) = \sum_{i=1}^m x_i g_i \in G$ and $c = \sum_{i=1}^m x_i y_i + \mu/2 \in \mathbb{T}_k$. Here, $\sum_{i=1}^m x_i y_i$ is Alice's key in the GLWE key exchange. Both d and c use the $\mathbb{Z}$-module structure of G and $\mathbb{T}_k$.

$\texttt{GLWE.Dec}(sk, (d, c))$: Returns $\mu = \lfloor 2 \cdot (c - \hat{s}(d)) \rceil \bmod 2$ where $sk = \hat{s}$ and $(d, c) \in H$ is the ciphertext.

One obtains a dual scheme by swapping the two one-way functions $f_{\mathbf{g}}$ and $f_{\mathbf{g}}^{\times}$: the secret key is $\mathbf{x} \in \mathbb{Z}^m \leftarrow \mathcal{D}$, and the public key is $y = f_{\mathbf{g}}(\mathbf{x})$. The ciphertext of a message $\mu \in \{0, 1\}$ is a pair $(d, \mathbf{c}) \in \mathbb{T} \times \mathbb{T}^m$ where $d = \mu/2 + \hat{s}(y) + e$, $\mathbf{c} = f_{\mathbf{g}}^{\times}(\hat{s}, \mathbf{e})$ where $(\hat{s}, \mathbf{e}, e)$ is a one-time key. Here, $\hat{s}(y) + e$ is the approximate shared key of the GLWE key exchange. A ciphertext $(d, \mathbf{c})$ is decrypted as $\lfloor 2(d - \sum_{i=1}^m c_i x_i) \rceil \bmod 2 \in \{0, 1\}$.

By analogy with El Gamal, breaking the semantic security of the main GLWE scheme is equivalent to solving the GLWE-DDH problem of distinguishing $(f_{\mathbf{g}}(\mathbf{x}), f_{\mathbf{g}}^{\times}(\hat{s}, \mathbf{e}), \sum_{i=1}^m c_i x_i)$ from $(f_{\mathbf{g}}(\mathbf{x}), f_{\mathbf{g}}^{\times}(\hat{s}, \mathbf{e}), c)$ where $c \in_R \mathbb{T}$, $\mathbf{x} \leftarrow \mathcal{D}$, $\hat{s} \in_R \hat{G}$, $\mathbf{e} \leftarrow \mathcal{D}_\alpha^m$ and $e \leftarrow \mathcal{D}$, and $\mathbf{c} = f_{\mathbf{g}}^{\times}(\hat{s}, \mathbf{e})$. However, unlike the DL setting, one can see that GLWE-DDH is equivalent to decisional-GLWE.

Lemma 7.1 (Correctness) *If* $0 < \alpha < 1/(4 \cdot \sqrt{m} \cdot \omega(\sqrt{\log n}))$, *then the main GLWE public-key encryption scheme will decrypt correctly with probability* $1 - negl(n)$.

Proof. We have: $c - \hat{s}(d) = \sum_{i=1}^m x_i(\hat{s}(g_i) + e_i) + \mu/2 - \hat{s}(\sum_{i=1}^m x_i g_i) = \mu/2 + \sum_{i=1}^m x_i e_i$. It is sufficient to show $|\sum_{i=1}^m x_i e_i| < 1/4$. Let $w \leq m$ be the Hamming weight of $\mathbf{x}$, we know that $\sum_{i=1}^m x_i e_i$ is distributed as $\mathcal{D}_{\sqrt{w}\alpha}$. Therefore, it implies that $|\sum_{i=1}^m x_i e_i| < \sqrt{w}\alpha \cdot \omega(\sqrt{\log n})$ with probability $1 - \exp(-\pi \cdot \omega(\log n)) = 1 - \mathrm{negl}(n)$. We obtain that $|\sum_{i=1}^m x_i . e_i| < \sqrt{w}\alpha \cdot \omega(\sqrt{\log n}) \leq 1/4$ with probability $1 - \mathrm{negl}(n)$, as desired. □

Lemma 7.2 (Security) *If* $m \geq 2(\log \#G + k) + \omega(\log n)$ *and the* $\mathrm{GLWE}_{G,m,\alpha}$ *assumption holds, then the main GLWE public-key encryption scheme is IND-CPA secure.*

Proof. $\mathbf{g} \in G^m$ is uniformly distributed. By the $\mathrm{GLWE}_{G,m,\alpha}$ assumption, $\mathbf{y} \in \mathbb{T}_k^m$ is computationally indistinguishable from uniform, hence $(\mathbf{g}, \mathbf{y})$ too. Since $m \geq 2 \cdot \log \#H + \omega(\log n)$ and $\mathbf{x} \in_R \{0, 1\}^m$, the left-over-hash lemma ensures that $\sum_{i=1}^m x_i(g_i, y_i)$ is computationally indistinguishable from uniform over H, and hence (d, c) too. This proves IND-CPA security. □

7.2 A GLWE Variant of GSW Homomorphic Encryption

We now show how to generalize the AP variant [3] of GSW [21] Homomorphic encryption. Let $\mathrm{GLWE}(G, \alpha)$ be a black-box instance of GLWE El Gamal encryption over the GLWE group G. All noises are discretized in the torus $\mathbb{T}_k = \frac{1}{2^k}\mathbb{Z}/\mathbb{Z} \subseteq \mathbb{T}$ where $2^k\alpha \approx \eta_\varepsilon(\mathbb{Z})$. The group $H = G \times \mathbb{T}_k$ is of special interest.

First, recall that El Gamal encryption is homomorphic with respect to the group operation. Because $\mathrm{GLWE}(G, \alpha)$ is a noisy variant of El Gamal encryption, it is also homomorphic for a bounded number of XOR. More precisely, any GLWE ciphertext of a message $\mu \in \{0,1\}$ can be written as $c_1 + \mu h_1 \in H$, where $c_1 = \sum_{i=1}^{m} x_i(g_i, y_i) \in H$ is a random ciphertext of 0, and $h_1 = (0, 1/2) \in H$. Here, we use the $\mathbb{Z}$-module structure of H. The GLWE secret key $\hat{s}$ induces a homomorphism $\texttt{Phase} : H \to \mathbb{T}$ defined as $\texttt{Phase}((a,b)) = b - \hat{s}(a)$. By definition of GLWE, we have $\texttt{Phase}((g_i, y_i)) \approx 0$ for all $1 \leq i \leq m$, but $\texttt{Phase}(h_1) = 1/2$. It follows that the phase of a GLWE ciphertext of a message μ is $\approx \mu/2$, which explains the GLWE decryption procedure: a ciphertext of 0 is close to the kernel of the phase, while a ciphertext of 1 is far away. Because $\texttt{Phase}$ is a homomorphism and h_1 has order 2 in H, if n messages $\mu_1, \ldots, \mu_n \in \{0,1\}$ are GLWE-encrypted, then the sum of these n ciphertexts will de decrypted as $\mu_1 \oplus \cdots \oplus \mu_n$, provided that n is not too large.

To achieve more homomorphic operations, one exploits a special property of lattice problems which is not shared by discrete logarithm problems: with special choices of generators, the SIS one-way function can be inverted. To do so, one first extends h_1 into a generating set of the $\mathbb{Z}$-module H: let $h_2, \ldots, h_\ell \in H$ be such that $\mathbf{h} = (h_1, \ldots, h_\ell)$ is a generating set of H. Recall that the GSIS function $f_{\mathbf{g}}$ from Sect. 7.1 can be defined over any group: here, we use H, so $f_{\mathbf{h}}(\mathbf{x}) = \sum_{i=1}^{\ell} x_i h_i \in H$ for $(x_1, \ldots, x_\ell) \in \mathbb{Z}^\ell$. Since $\mathbf{h}$ generates H, $f_{\mathbf{h}}$ is surjective, and thus, admits a pseudo-inverse $f_{\mathbf{h}}^{-1}$ from H to $\mathbb{Z}^\ell$, such that $f_{\mathbf{h}}(f_{\mathbf{h}}^{-1}(b)) = b$ for any $b \in H$. We also define $F_{\mathbf{h}} : \mathbb{Z}^{\ell \times \ell} \to H^\ell$ by $F_{\mathbf{h}}(\mathbf{X}) = (f_{\mathbf{h}}(\mathbf{x}_1), ..., f_{\mathbf{h}}(\mathbf{x}_\ell))$, where $\mathbf{x}_i$ is the i-th row of $\mathbf{X}$. Accordingly, we define $F_{\mathbf{h}}^{-1} : H^\ell \to \mathbb{Z}^{\ell \times \ell}$.

Given a target in H, finding a short $f_{\mathbf{h}}()$-preimage corresponds to the GSIS problem, which is in general hard, but it becomes easy for special choices of $\mathbf{h}$, like super-increasing knapsacks: following [26], we call *gadget* such a $\mathbf{h}$. We say that $f_{\mathbf{h}}^{-1}()$ is β-bounded for $\mathbf{h}$, if $\left\| f_{\mathbf{h}}^{-1}(b) \right\|_\infty \leq \beta \in \mathbb{R}^+$ for any $b \in H$. For instance, if the group G is $\mathbb{Z}_N$ where $2^p < N < 2^{p+1}$, a suitable gadget is $\mathbf{h} = ((0, \frac{1}{2}), (0, \frac{1}{4}), \ldots, (0, \frac{1}{2^k}), (1,0), (2,0), \ldots, (2^p, 0))$, $f_{\mathbf{h}}^{-1}() \in \{0,1\}^\ell$ can be computed by binary decomposition and is 1-bounded for $\mathbf{h}$. This construction can easily be generalized to any fully-explicit G, using component-wise binary decomposition: if $G = \mathbb{Z}_q^n$, this corresponds to the Flatten/BitDecomp algorithms proposed in [21] and [3]. However, other algorithms are possible, such as ternary decompositions with preimages in $\{0, \pm 1\}^\ell$.

Given the GLWE encryption scheme $(\texttt{GLWE.Gen}, \texttt{GLWE.Enc}, \texttt{GLWE.Dec})$ described in Sect. 7.1 as a "black box", we build homomorphic encryption using a gadget $\mathbf{h} \in H^\ell$ whose first element is $(0, \frac{1}{2})$:

$\texttt{GSW.Gen}(1^n)$: Takes as input a security parameter n, it runs the key generation algorithm $(pk, sk) \leftarrow \texttt{Gen}(1^n)$, where $pk = (\mathbf{g}, \mathbf{y}) \in G^m \times \mathbb{T}_k^m$ and $sk = \hat{s} \in \hat{G}$.

$\texttt{GSW.Enc}(pk, \mu)$: Takes as input the public key $pk \in G^m \times \mathbb{T}_k^m$ and a message $\mu \in \{0,1\}$, it first generates ℓ ciphertexts $c_1 = \texttt{GLWE.Enc}(pk, 0), ..., c_\ell = \texttt{GLWE.Enc}(pk, 0)$ of zero, and returns

$$\mathbf{c} = (c_1, ..., c_\ell) + \mu \cdot \mathbf{h} \in H^\ell.$$

This is reminiscent of the GLWE scheme, where a GLWE-ciphertext of a message μ is of the form $c_1 + \mu h_1 \in H$ where c_1 is a random GLWE-ciphertext of 0. Because the first entry of $\mathbf{h}$ is $(0, \frac{1}{2})$, the first entry of $\mathbf{c}$ is a GLWE encryption of μ.

$\texttt{GSW.Dec}(sk, \mathbf{c})$: Returns $\texttt{GLWE.Dec}(\hat{s}, c_1)$ where $sk = \hat{s}$ and $c_1 \in H$ is the first entry of $\mathbf{c}$.

The security of the scheme and the correctness of decryption follow from that of the GLWE cryptosystem:

Lemma 7.3 *Suppose* $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ *uses samples from* $GLWE_{G,m,\alpha}$. *If* $m \geq 2(\log \#G + k) + \omega(\log n)$ *and* $0 < \alpha < 1/(4 \cdot \sqrt{m} \cdot \omega(\sqrt{\log n}))$, $(\texttt{GSW.Gen}, \texttt{GSW.Enc}, \texttt{GSW.Dec})$ *is IND-CPA secure under the* $GLWE_{G,m,\alpha}$ *assumption, and* $\texttt{GSW.Dec}$ *decrypts correctly with probability* $1 - \texttt{negl}(\lambda)$.

Proof. The proof of IND-CPA security is similar to Lemma 7.2. Since the first entry of $\mathbf{c}$ is a ciphertext of μ under $\hat{s}$ of the scheme $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$, correctness follows from Lemma 7.1. $\square$

We now describe our homomorphic operations on ciphertexts, namely how to encode Not, And, and Mux gates. First, we note that the GSW-GLWE scheme inherits the $\oplus$-homomorphic properties of the GLWE scheme. It is classical that any circuit can be built using only Not and And elementary gates. We chose to add the Mux ternary gate, which encodes the conditional operator $\texttt{Mux}(a, b, c) = a?b{:}c$, because resulting circuits are smaller than NAND-only circuits, all binary gates can be encoded by a single Mux (and a few Not), and it is trivial to batch-convert any truth-table to its corresponding Mux-based binary decision diagram.

Definition 7.4 (Homomorphic operations) *For all ciphertexts $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3 \in H^\ell$, we define:*

$$\begin{aligned}\texttt{GSW.Not}(\mathbf{c}_1) &= \mathbf{h} - \mathbf{c}_1,\\ \texttt{GSW.And}(\mathbf{c}_1, \mathbf{c}_2) &= F_{\mathbf{c}_1}\Big(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)\Big),\\ \texttt{GSW.Mux}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) &= F_{\mathbf{c}_1}\Big(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)\Big) + F_{\mathbf{h}-\mathbf{c}_1}\Big(F_{\mathbf{h}}^{-1}(\mathbf{c}_3)\Big)\end{aligned}$$

We express $\texttt{Xor}(a, b)$ as $\texttt{Mux}(a, \texttt{Not}(b), b)$.

We naturally extend the Phase homomorphism to H^ℓ as $\texttt{Phase} : H^\ell \to \mathbb{T}^\ell$ defined as $\texttt{Phase}(\mathbf{z}) = (b_1 - \hat{s}(a_1), \dots, b_\ell - \hat{s}(a_\ell)) \in \mathbb{T}^\ell$ where $\mathbf{z} = ((a_1, b_1), \dots, (a_\ell, b_\ell)) \in H^\ell$. Note that a valid ciphertext of a bit μ is of the form $\mathbf{c} = \mathbf{z} + \mu\mathbf{h}$ where its *homogeneous* part $\mathbf{z}$ has a small phase. This small $\texttt{Phase}(\mathbf{z}) = \texttt{Phase}(\mathbf{c} - \texttt{GSW.Dec}(\mathbf{c}).\mathbf{h}) \in \mathbb{T}^\ell$ will be denoted by $\texttt{Noise}(\mathbf{c})$.

By definition, the decryption function will successfully decrypt any ciphertext $\mathbf{c} \in H^\ell$ such that $\|\texttt{Noise}(\mathbf{c})\|_\infty < \frac{1}{4}$, where the max-norm in $\mathbb{T}^\ell$ is taken over all coordinates centered in the interval $(-\frac{1}{2}, \frac{1}{2}]$. This is of course the case of fresh $\texttt{GSW.GLWE}$ ciphertexts, whose Gaussian noise has small parameter α.

We now show that the $\texttt{GSW.Not}$, $\texttt{GSW.And}$ and $\texttt{GSW.Mux}$ gates amplify the noise only by a small factor if $f_{\mathbf{h}}^{-1}()$ is β-bounded.

Lemma 7.5 (Worst-case noise of primitive gates) *Suppose $f_{\mathbf{h}}^{-1}()$ is β-bounded for some $\beta \in \mathbb{R}^+$. Let $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3 \in H^\ell$ be three ciphertexts such that $\mathbf{c}_1 = \mathbf{z}_1 + \mu_1 \cdot \mathbf{h}$, $\mathbf{c}_2 = \mathbf{z}_2 + \mu_2 \cdot \mathbf{h}$ and $\mathbf{c}_3 = \mathbf{z}_3 + \mu_3 \cdot \mathbf{h}$, where $\|\texttt{Phase}(\mathbf{z}_1)\|_\infty \le B$ and $\|\texttt{Phase}(\mathbf{z}_2)\|_\infty, \|\texttt{Phase}(\mathbf{z}_3)\|_\infty < B'$ for some $B, B' \in \mathbb{R}^+$. Then:*

$$\texttt{GSW.Not}(\mathbf{c}_1) = \mathbf{z} + \texttt{NOT}(\mu_1) \cdot \mathbf{h} \text{ where } \|\texttt{Phase}(\mathbf{z})\|_\infty = B \tag{4}$$

$$\texttt{GSW.And}(\mathbf{c}_1, \mathbf{c}_2) = \mathbf{z}' + (\mu_1 \texttt{ AND } \mu_2) \cdot \mathbf{h} \text{ where } \left\|\texttt{Phase}(\mathbf{z}')\right\|_\infty \le \ell\beta B + B' \tag{5}$$

$$\texttt{GSW.Mux}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) = \mathbf{z}'' + (\mu_1?\mu_2{:}\mu_3) \cdot \mathbf{h} \text{ where } \left\|\texttt{Phase}(\mathbf{z}'')\right\|_\infty \le 2\ell\beta B + B' \tag{6}$$

Proof.

By definition of GSW, we have $\texttt{GSW.Not}(\mathbf{c}_1) = -\mathbf{z}_1 + \texttt{NOT}(\mu_1)$, so $\mathbf{z} = -\mathbf{z}_1$, which proves (4). Then,

$$\begin{aligned}\texttt{GSW.And}(\mathbf{c}_1, \mathbf{c}_2) &= F_{\mathbf{c}_1}\Big(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)\Big) = F_{\mathbf{z}_1+\mu_1\cdot\mathbf{h}}\Big(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)\Big) = F_{\mathbf{z}_1}(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)) + \mu_1 F_{\mathbf{h}}(F_{\mathbf{h}}^{-1}(\mathbf{c}_2))\\ &= F_{\mathbf{z}_1}(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)) + \mu_1 \cdot \mathbf{c}_2 = \underbrace{F_{\mathbf{z}_1}(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)) + \mu_1\mathbf{z}_2}_{\mathbf{z}'} + \mu_1\mu_2 \cdot \mathbf{h}\end{aligned}$$

Letting $\mathbf{z}' = F_{\mathbf{z}_1}(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)) + \mu_1\mathbf{z}_2$, we have $\texttt{Phase}(\mathbf{z}') = \texttt{Phase}(\mathbf{z}_1) \cdot (F_{\mathbf{h}}^{-1}(\mathbf{c}_2))^t + \mu_1\texttt{Phase}(\mathbf{z}_2)$, and therefore $\|\texttt{Phase}(\mathbf{z}')\|_\infty \le \ell \left\|F_{\mathbf{h}}^{-1}(\mathbf{c}_2)\right\|_\infty \|\texttt{Phase}(\mathbf{z}_1)\|_\infty + \|\texttt{Phase}(\mathbf{z}_2)\|_\infty \le \ell\beta B + B'$, which proves (5). Finally, $\texttt{GSW.Mux}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ is expressed as $\texttt{GSW.And}(\mathbf{c}_1, \mathbf{c}_2)$ plus

$\mathtt{GSW.And}(\mathtt{GSW.Not}(\mathbf{c}_1), \mathbf{c}_3)$. By expanding, the expression takes the form $\mathbf{z}''+(\mu_2\mu_1+\mu_3(1-\mu_1))\cdot\mathbf{h}$ where $\mathbf{z}'' = F_{\mathbf{z}_1}(F_{\mathbf{h}}^{-1}(\mathbf{c}_2)) + F_{\mathbf{z}_1}(F_{\mathbf{h}}^{-1}(\mathbf{c}_3)) + \mu_1\mathbf{z}_2 + (1-\mu_1)\mathbf{z}_3$. Thus, $\mathtt{Phase}(\mathbf{z}'') = \mathtt{Phase}(\mathbf{z}_1)\cdot(F_{\mathbf{h}}^{-1}(\mathbf{c}_2) + F_{\mathbf{h}}^{-1}(\mathbf{c}_3)) + \mu_1\mathtt{Phase}(\mathbf{z}_2) + (1-\mu_1)\mathtt{Phase}(\mathbf{z}_3)$. The norm of the first term is bounded by $2\ell\beta B$ and among the last two terms, only one is non-zero, and its norm is bounded by B'. Finally, the encoded message $\mu_2\mu_1 + \mu_3(1-\mu_1)$ is precisely $\mu_1?\mu_2{:}\mu_3$. □

In the next section, we will use the worst-case recurrence equations (4), (5) and (6) to upper-bound the noise of more complex circuits. Before that, we want to point out that similarily to [3], if $f_{\mathbf{h}}^{-1}$ satisfies a few additional constraints, we can prove a very natural result: if the input ciphertexts have independent Gaussian noise, we can ensure that all other ciphertexts which are produced by the homomorphic gates also have independent Gaussian noise, and the square parameter of the noise follows the same recurrence as the worst-case max-norm. Thus, with overwhelming probability, the actual noise norm is in fact the square root of what can be predicted by the worst-case bound from Lemma 7.5.

We say that $f_{\mathbf{h}}^{-1}$ is β-Gaussian if for each $y \in H$, $f_h^{-1}(y)$ returns a discrete Gaussian sample of the coset $\{\mathbf{x} \in \mathbb{Z}^\ell \text{ s.t. } f_h(\mathbf{x}) = y\}$ centered in 0 and of parameter $\beta \geq \eta_\varepsilon(\mathcal{L}_{\mathbf{h}})$.

Lemma 7.6 (all noises are Gaussian) *Suppose that $f_{\mathbf{h}}^{-1}$ is β-Gaussian for $\beta \geq \eta_\varepsilon(\mathcal{L}_{\mathbf{h}})$. In a circuit containing solely* GSW.Not*,* GSW.And *and* GSW.Mux *gates, and whose inputs are either fresh GLWE ciphertexts or the noiseless ciphertexts* 0 *and* $\mathbf{h}$*, the output ciphertext of each individual gate has the form $\mathbf{z}+\mu\mathbf{h}$ where μ is the encoded bit and the ℓ-coordinates of* $\mathtt{Phase}(\mathbf{z})$ *are indistinguishable from independent Gaussian samples of $\mathbb{T}_k$. We define the noise parameter $\sigma(\mathtt{Phase}(z))$ as the maximum of these ℓ Gaussian parameters.*

Proof. (sketch) It suffices to verify by induction on the depth of the circuit for the three gates that the noise $\mathtt{Phase}(z)$ of the output ciphertext could be decomposed as $\sum_{i=1}^{N}\mathtt{Phase}(\mathbf{z}_i)(\sum_{j=1}^{N_i} \pm \prod_{k=1}^{j} F_{\mathbf{h}}^{-1}(c_{i,j,k}) + \alpha_i I_\ell)^t$ where N is bounded by the number of inputs, $N_1, \dots, N_N \in \mathbb{N}$ are bounded by the depth of the ciphertext in the circuit, where all $\mathbf{c}_{i,j,k}$ are (possibly equal) ciphertexts of the circuit, $\alpha_i \in \{-1,0,1\}$ and where $(\mathbf{z}_1, \dots, \mathbf{z}_N)$ are the homogeneous parts of different input ciphertexts. Since the rows of each $F_{\mathbf{h}}^{-1}(c_{i,j,k})$ have independent zero-centered discrete Gaussian distributions on their respective domains, the rows of their their product are also independent discrete Gaussian samples. The rows of the sum of such products are also independent discrete Gaussian samples. Since all the coordinates of the input $\mathtt{Phase}(\mathbf{z}_i)$ are independent zero-centered Gaussian samples of $\mathbb{T}_k$, so is their total combination, up to some negligible statistical distance. □

Note that the noise parameter of a fresh GSW.GLWE ciphertext satisfies $\sigma(\mathtt{Phase}(z)) \leq \alpha$ where α is the underlying GLWE Gaussian parameter. Gaussian parameters follow Pythagorean summations: a linear combination of independent zero-centered Gaussian samples is a zero-centered Gaussian sample whose parameter is multiplied by the Euclidean norm of the combination[2]. This leads to a tighter average-case version of Lemma 7.5.

Lemma 7.7 (Average noise of primitive gates) *Suppose $f_{\mathbf{h}}^{-1}()$ is $\sqrt{\beta}$-Gaussian for some $\beta \geq (\eta_\varepsilon(\mathcal{L}_{\mathbf{h}}))^2$. Let $\mathbf{c}_1 = \mathbf{z}_1 + \mu_1\cdot\mathbf{h}$, $\mathbf{c}_2 = \mathbf{z}_2 + \mu_2\cdot\mathbf{h}$ and $\mathbf{c}_3 = \mathbf{z}_3 + \mu_3\cdot\mathbf{h} \in H^\ell$ be three ciphertexts of a circuit satisfying the constraints of Lemma 7.6, and whose Gaussian parameters*

[2] This is always true for continuous gaussian samples. For discrete ones, it remains true when all the parameters are larger than the smoothing parameter of the discretization subgroup, and when the coefficients of the linear combination are relatively prime (see Theorem 3 of [27]).

satisfy $\sigma(\texttt{Phase}(\mathbf{z}_1))^2 \leq B$ *and* $\sigma(\texttt{Phase}(\mathbf{z}_2))^2, \sigma(\texttt{Phase}(\mathbf{z}_3))^2 < B'$ *for some* $B, B' \in \mathbb{R}^+$*. Then:*

$$\texttt{GSW.Not}(\mathbf{c}_1) = \mathbf{z} + \texttt{NOT}(\mu_1) \cdot \mathbf{h} \textit{ where } \sigma(\texttt{Phase}(\mathbf{z}))^2 = B \tag{7}$$

$$\texttt{GSW.And}(\mathbf{c}_1, \mathbf{c}_2) = \mathbf{z}' + (\mu_1 \texttt{ AND } \mu_2) \cdot \mathbf{h} \textit{ where } \sigma(\texttt{Phase}(\mathbf{z}'))^2 \leq \ell\beta B + B' \tag{8}$$

$$\texttt{GSW.Mux}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) = \mathbf{z}'' + (\mu_1 ? \mu_2 : \mu_3) \cdot \mathbf{h} \textit{ where } \sigma(\texttt{Phase}(\mathbf{z}''))^2 \leq 2\ell\beta B + B' \tag{9}$$

Proof. We saw in the proof of Lemma 7.5 that the noise of $\texttt{GSW.And}(\mathbf{c}_1, \mathbf{c}_2)$ is $\texttt{Phase}(\mathbf{z}') = \texttt{Phase}(\mathbf{z}_1) \cdot (F_{\mathbf{h}}^{-1}(\mathbf{c}_2))^t + \mu_1 \texttt{Phase}(\mathbf{z}_2)$. Since $F_{\mathbf{h}}^{-1}(\mathbf{c}_2)$ and $\texttt{Phase}(\mathbf{z}_2)$ are independent Gaussian samples on their respective domains, and $\texttt{Phase}(\mathbf{z}_1)$, $\texttt{Phase}(\mathbf{z}_2)$ are zero-centered, then the Gaussian parameters follow Pythagorean summation. Thus, $\sigma(\texttt{Phase}(\mathbf{z}'))^2 \leq \ell\beta B + B'$ which proves (8). We prove (9) similarly. □

Since the recurrences (7), (8) and (9) on the square noise parameter are exactly the same as the recurrences (4), (5) and (6) on the max-norm, we will continue the analysis only on the max-norm, whose requirements on $f_{\mathbf{h}}^{-1}$ and on the definition of the probability space are simpler to express. The reader has to keep in mind that all the bounds we deduce on the max-norm of the noise also applies to its square parameter.

7.3 Homomorphically Evaluating Arbitrary Functions

The result of the following corollary was already obtained in [3]; it states that in a long chain of And gates where one of the bits is a fresh GLWE-GSW ciphertext, the noise increases in fact linearly instead of exponentially. Here, we invert the operands of the And gates, so the overall noise in the resulting ciphertext is smaller if one associates long conjunctions on the right.

Corollary 7.8 (Noise of Conjunctions) *Suppose* $f_{\mathbf{h}}^{-1}()$ *is* β*-bounded for some* $\beta \in \mathbb{R}^+$*. Let* $\mathbf{c}_1, \ldots, \mathbf{c}_k \in H^\ell$ *be* k *ciphertexts such that each* $\mathbf{c}_i = \mathbf{z}_i + \mu_i \cdot \mathbf{h}$ *where* $\|\texttt{Phase}(\mathbf{z}_i)\|_\infty < B$ *for some* $B \in \mathbb{R}^+$*. Then:*

$$\texttt{GSW.And}(\mathbf{c}_1, \texttt{GSW.And}(\mathbf{c}_2, \ldots \texttt{GSW.And}(\mathbf{c}_{k-1}, \mathbf{c}_k))) = \mathbf{z} + (\mu_1\mu_2 \ldots \mu_k) \cdot \mathbf{h} \textit{ where } \|\texttt{Phase}(\mathbf{z}\|_\infty \leq k\ell\beta B \tag{10}$$

Proof. Apply (5) by induction on k. □

Note that any boolean function with k inputs can always be put into disjunctive normal form, *i.e.* a disjoint union of conjunctive terms, and one way to homomorphically evaluate the result is to add the ciphertexts of all the terms, which indeed preserves the $\{0, 1\}$ message space. However, using this method, the resulting noise will be proportional to the number of terms in the disjunctive normal form, which may still be exponential in the number of inputs.

By using Mux-gates, we obtain the following corollary, which truly reflects the homomorphic nature of the cryptosystem. It basically says that any function can be homomorphically evaluated in a trivial way, where the noise grows proportionally to only the square root of the number of inputs. We recall that the truth table of a boolean function ϕ with k variables is a vector $\mathcal{T}$ of length 2^k such that each $\mathcal{T}_j = \phi(e_0, \ldots, e_{k-1})$ where $j = \sum e_i 2^{k-1-i}$. The full binary decision diagram (BDD) of ϕ is a circuit representing a binary tree of Mux-gates, of depth k. The bottom level k consists in 2^k leaves $X_{k,j}$, each one is set to $\mathcal{T}_j$. At each intermediate level i, we have 2^i nodes $X_{k,j} = \texttt{Mux}(\mu_i, X_{i+1,2j+1}, X_{i+1,2j})$. By definition, the root $X_{0,0}$ thus contains $\phi(\mu_0, \ldots, \mu_{k-1})$. See Fig. 2 for an example of truth table and its associated BDD circuit.

Corollary 7.9 (Evaluating arbitrary functions) *Assume that* $f_{\mathbf{h}}^{-1}()$ *is* β*-bounded for some* $\beta \in \mathbb{R}^+$*. Let* ϕ *be any boolean function with* k *inputs, and let* $\mathbf{c}_1, \ldots, \mathbf{c}_k \in H^\ell$ *be* k *ciphertexts such that each* $\mathbf{c}_i = \mathbf{z}_i + \mu_i \cdot \mathbf{h}$ *where* $\sigma(\mathbf{z}_i)^2 < B$ *for some* $B \in \mathbb{R}^+$*. Then, the Mux-based Binary Decision Diagram of* ϕ *computes a ciphertext* $\mathbf{c} = \mathbf{z} + \phi(\mu_1, \ldots, \mu_k).\mathbf{h}$ *where* $\|\mathbf{z}\|_\infty \leq 2k\ell\beta B$*.*

Proof. To evaluate the full BDD of ϕ homomorphically, we just replace each leaf $X_{k,j}$ by noiseless ciphertexts $\mathcal{T}_j.\mathbf{h}$, each bit μ_i by their encryption $\mathbf{c_i}$, and each Mux gate by GSW.Mux. Apply (6) by induction on the depth, then all nodes $X_{i,j}$ at depth i have a noise bounded by $2(k-i)\beta B$. □

In the previous corollary, the full BDD tree of the function ϕ contains a number of nodes which is exponential in the number of inputs. If the output noise is indeed really small, the time complexity to evaluate all the gates remains large when the simulated function has many variables. For some useful functions, like the bootstrapping function in the next section, many of the subtrees turn out to be equal, and by merging them, the complexity to evaluate the circuit can be significantly reduced.

Corollary 7.10 (Faster Evaluation of arbitrary functions) *Suppose $f_{\mathbf{h}}^{-1}()$ is β-bounded for some $\beta \in \mathbb{R}^+$. Let ϕ be any boolean function with k inputs, and let $\mathbf{c}_1, \ldots, \mathbf{c}_k \in H^\ell$ be k ciphertexts such that each $\mathbf{c}_i = \mathbf{z}_i + \mu_i \cdot \mathbf{h}$ where $\|\mathtt{Phase}(\mathbf{z}_i)\|_\infty < B$ for some $B \in \mathbb{R}^+$. We call $\mathcal{N}(\phi)$ the number of disctinct subtrees in the full Binary Decision Diagram of ϕ. Then, we can compute a ciphertext $\mathbf{c} = \mathbf{z} + \phi(\mu_1, \ldots, \mu_k).\mathbf{h}$ where $\|\mathtt{Phase}(\mathbf{z})\|_\infty \leq 2k\ell\beta B$ by evaluating $\mathcal{N}(\phi)$ homomorphic* GSW.Mux*-gates.*

Proof. It suffices to evaluate the ciphertext value in the root of the $\mathcal{N}(\phi)$ subtrees by increasing depth. There are at most two different leaves, whose ciphertext values 0 and $\mathbf{h}$ are given. Whenever we need to evaluate a subtree of non zero depth i, the left and right subtrees have by definition already been fully evaluated, since their depth $i-1$ is strictly smaller. The root of the current tree is the GSW.Mux of $\mathbf{c}_i$ and the two subtrees roots. The last ciphertext to be evaluated is the root of the full tree, which contains the encrypted result. □

Note that in the above corollary, Nerode's partitioning algorithm for reducing deterministic automata can efficiently list the $\mathcal{N}(\phi)$ identical subtrees. It is actually not so surprising that algorithms from automata theory appear to be useful here, since a binary decision diagram is just the mirror graph of a deterministic accessible automata. Note that the GSW.Mux gate is all we need to efficiently homomorphically evaluate the transitions of a deterministic automata, which leads to the following lemma.

Lemma 7.11 (Recognizing arbitrary rational langages) *Let $\mathcal{L}$ be an arbitrary rational language of $\{0,1\}^*$ and $\mathcal{N}(\tilde{\mathcal{L}})$ the number of residuals of the mirror language of $\mathcal{L}$. Given k ciphertexts $\mathbf{c}_1, \ldots, \mathbf{c}_k$ of a message $\mathbf{w} = w_1, \ldots, w_k$, one can compute a ciphertext $\mathbf{c} = \mathbf{z} + \mathcal{L}(\mathbf{w}).\mathbf{h}$ where $\mathcal{L}(\mathbf{w}) = 1$ iff $\mathbf{w} \in \mathcal{L}$ and $\|\mathtt{Phase}(\mathbf{z})\|_\infty \leq 2k\beta B$ by evaluating $k\mathcal{N}(\tilde{\mathcal{L}})$* GSW.Mux*-gates.*

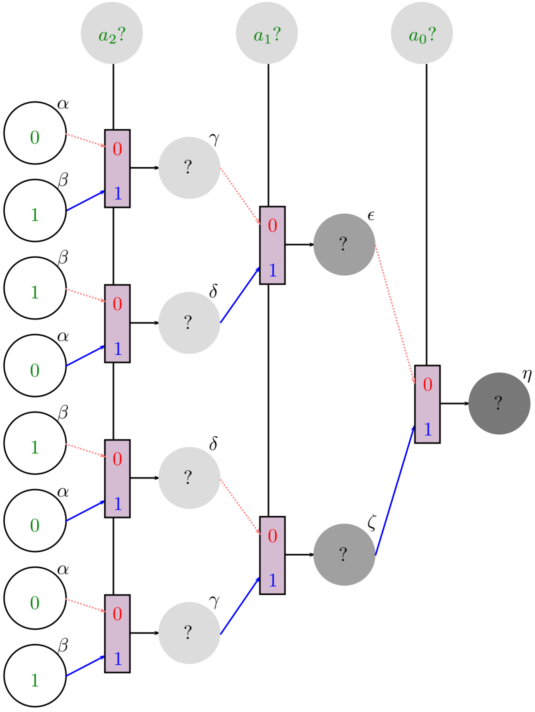
Proof. Let $\mathcal{A} = (Q, i, T_0, T_1, F)$ be a minimal deterministic automata of the mirror language $\tilde{\mathcal{L}}$ where Q is the set of states, $i \in Q$ is the initial state, T_0, T_1 are the two transitions functions from Q to Q and F is the set of final states. Note that $\#Q = \mathcal{N}(\tilde{\mathcal{L}})$. We initialize $\#Q$ noiseless ciphertexts $X_{q,0}$ for $q \in Q$ with $X_{q,0} = \mathbf{h}$ if $q \in F$ and $X_{q,0} = 0$ otherwise. Then for each letter we compute the transition as follow: $X_{q,j} = \mathtt{GSW.Mux}(c_j, X_{T_1(q),j-1}, X_{T_0(q),j-1})$. And we output $X_{i,k}$. We write $\mathbf{a} \equiv \mathbf{b}$ when two ciphertexts $\mathbf{a}$ and $\mathbf{b} \in H^\ell$ encrypt the same bit. Then we have $X_{i,k} \equiv X_{T_{w_k}(i),k-1} \equiv \ldots \equiv X_{T_{w_1}(T_{w_2}\cdots(T_{w_k}(i))\ldots),0}$, which encrypts 1 iff $T_{w_1}(T_{w_2} \ldots (T_{w_k}(i)) \ldots) \in F$, *i.e.* iff $w_k \ldots w_1$ is accepted by $\mathcal{A}$ iff $w_1 \ldots w_k \in \mathcal{L}$. This proves correctness.

For the complexity, each $X_{q,j}$ is computed with a single GSW.Mux gate and the noise increases as in the previous corollary since the fresh-GSW.Mux depth of the circuit is k. □

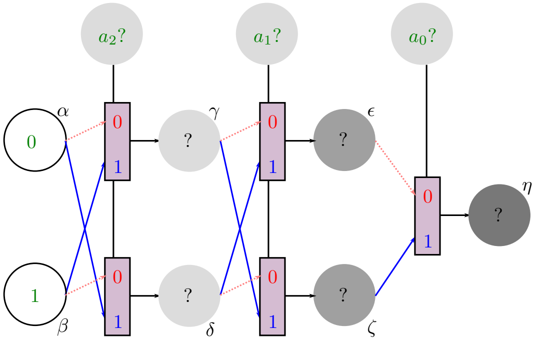
It remains an open question whether one can extend these lemmata to non-deterministic automata, or to Turing machines. Although they correspond to simple Mux-based circuits, the main problem is that the control bit of the Mux gates would not be a fresh ciphertext

a_0	a_1	a_2	ϕ
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Truth table of ϕ



Full BDD of ϕ (Corollary 7.7)



Reduced BDD of ϕ (Corollary 7.8)

Figure 2: Homomorphic evaluation of an arbitrary boolean function

Any truth table (top left) can be batch-converted to a full BDD (Binary decision diagram) circuit (top right). When evaluated homomorphically, the output noise is proportional to the square root of the number of inputs. After merging identical subtrees, which can be efficiently achieved using Nerode's partitioning algorithm, one obtains the reduced BDD circuit (bottom right), which may sometimes be much smaller. Here, the function ϕ was the Xor of all inputs. In this case, there are only two different subtrees per levels, and the reduced BDD is therefore polynomial in the number of inputs.

anymore, and thus, the noise would grow exponentially instead of sub-linearly, which doesn't seem suitable for useful applications. Luckily, it appears that the decoding function is a simple arithmetic circuit, and that a direct application of Corollary 7.9 suffices to bootstrap the whole system, turning it into a fully homomorphic one.

7.4 Simple Bootstrapping Circuit with Polynomial Noise

In this section, we present a very simple and generic decryption procedure, which makes the above GLWE-GSW fully-homomorphic under the GLWE assumption with inverse polynomial Gaussian parameter. To ease the presentation, we suppose that the gadget $\mathbf{h}$ of the previous section corresponds to the binary decomposition, and thus the image of $f_{\mathbf{h}}^{-1}$ is $\{0,1\}^{\ell}$, which is 1-bounded.

Recall that bootstrapping refers to Gentry's homomorphic decryption, which allows to transform suitable somewhat-homomorphic schemes into fully-homomorphic schemes. In our case, the decryption procedure is simply GLWE decryption. The idea is very simple, to decrypt a GLWE ciphertext $c \in H$, or more precisely, its decomposition $\mathbf{x} = f_{\mathbf{h}}^{-1}(c) \in \{0,1\}^{\ell}$, it suffices to evaluate the phase $\texttt{Phase}(c) = \sum_{i=1}^{\ell} x_i \texttt{Phase}(h_i) \in \mathbb{T}$ and decide whether it is closer to 0 or 1/2. Moreover, note that only $\log_2(\ell)+3$ bits (we assume ℓ is power of 2 for simplicity) of precision in each $\texttt{Phase}(h_i)$ are needed to decrypt correctly, as the contribution of all remaining bits cannot affect the most significant bit in a sum of ℓ elements. This means that for bootstrapping, it is enough be able to add ℓ numbers modulo 8ℓ.

Since the $\texttt{Phase}$ function involves the private key, all we need to do is to provide, as the bootstrapping key, fresh encryptions of the first $n = \log_2(\ell)+3$ bits of each $\texttt{Phase}(h_1), \ldots, \texttt{Phase}(h_\ell)$.

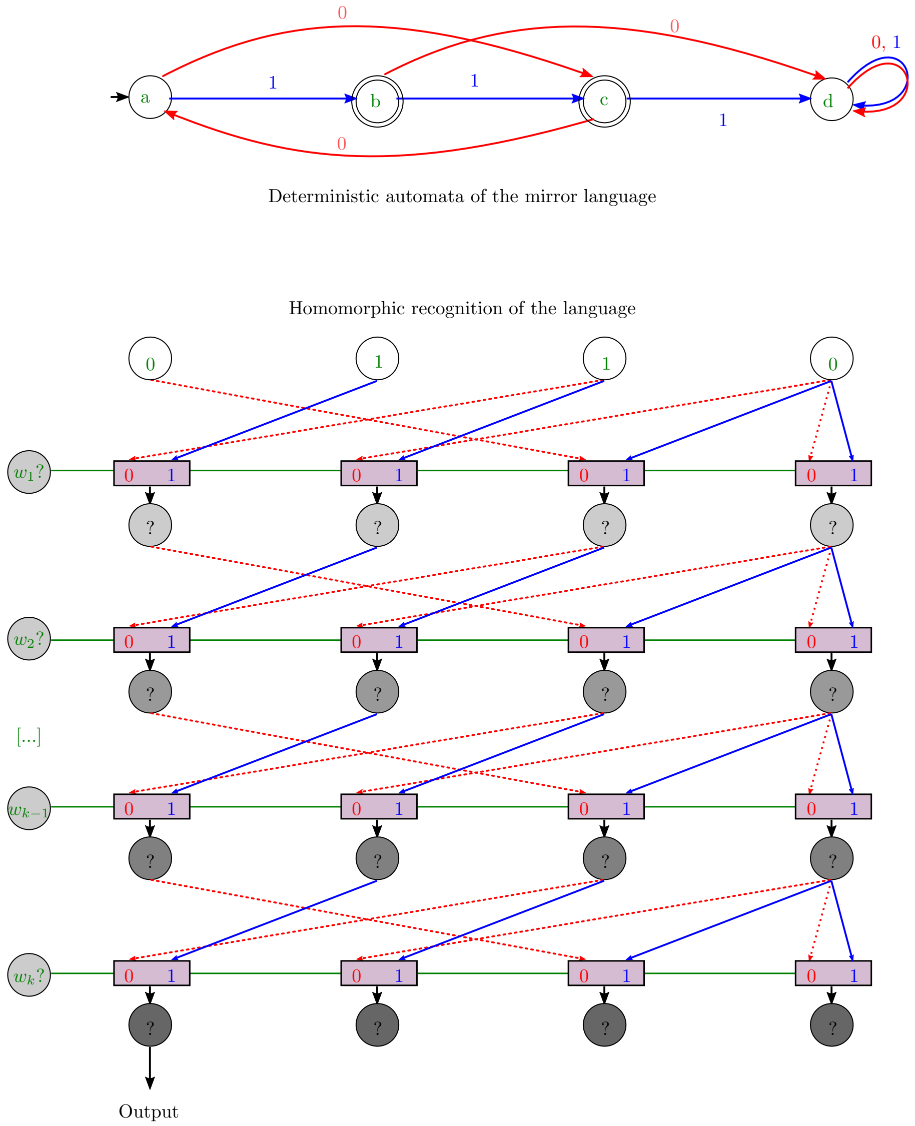
Note that decryption is a boolean function ϕ with ℓn input bits, so Corollary 7.9 already provides a trivial decryption circuit whose noise parameter expansion is only $O(\ell\sqrt{n})$. In fact, a careful analysis shows that since we are evaluating a simple arithmetic circuit, each level of the BDD tree contains at most 8ℓ identical sub-trees, each one corresponding to a different accumulated value modulo 8ℓ. Thus, the number $\mathcal{N}(\phi)$ of different subtrees is quadratic in ℓ, and Corollary 7.10 evaluates it in time $\tilde{O}(\ell^2)$.

We propose a simpler and equivalent view of the decryption circuit. Suppose that we have a state of 8ℓ boxes (or ciphertexts) arranged along a circle. We define adding v to the state as circularly shifting the state by v positions to the right. Let $\mathbf{b}$ be an encrypted bit, adding $2^k\mathbf{b}$ to the state corresponds to the following operation: if $\mathbf{b} \overset{?}{=} 0$, then each box keeps its previous content, else each box gets the previous content of the box 2^k positions on the left. This operation can thus be done homomorphically with one $\texttt{GSW.Mux}$-gate per box. Adding an encrypted n-bit number to the state corresponds to performing this operation successively for the n encrypted bits, each one associated with its corresponding power of 2. Finally, in order to decide whether the total rotation we applied was closer to 4ℓ than to 0, it suffices to initialize all the boxes at positions 2ℓ to 6ℓ with a ciphertext of 1, all others boxes are initialized to 0, and in the end, measuring the box at position 0 gives the result. The corresponding bootstrapping algorithm is summarized in Alg. 6 and in Figure 4.

There are three important differences with [3].

- The first one is that the large cycle is decomposed using the CRT into smaller ones of prime orders, which on one hand, makes the circuit size quasi-linear instead of quadratic, but which includes a decoding phase on non-fresh ciphertexts before the final rounding. This increases their noise in the final ciphertext by some polynomial factor. (See Lemma 7.13 below for an alternative CRT circuit).

- The second difference is that [3] does not use Mux-gates: they map a cycle into equivalent



Deterministic automata of the mirror language

Homomorphic recognition of the language

Figure 3: A Mux-based circuit recognizing a regular language $\mathcal{L}$ and the corresponding deterministic automata of the mirror $\tilde{\mathcal{L}}$.

The automata on the top recognizes the regular expression $\tilde{\mathcal{L}}$ =(110|00)*(1|11|0). The circuit below recognizes its mirror $\mathcal{L}$ =((110|00)*(11|0))|((101)*1). The circuit transcripts the mirror graph of the automata: final states b, c are initialized with a noiseless ciphertext of 1, non-final states a, d with 0. All transitions are reversed and point to a `GSW.Mux` controlled by the encrypted current letter. After processing all letters, the final output is the ciphertext corresponding to the automata's initial state a. As explained in the proof of Lemma 7.11, The result encrypts one iff the word is in $\mathcal{L}$.

Algorithm 6 Bootstrapping algorithm

```
Input: A GLWE ciphertext $c \in H$, the gadget $\mathbf{h}$ and its functions $f_{\mathbf{h}}^{-1}$, and the bootstrapping
    key $(BK_{i,j})_{i\in[1,\ell],j\in[1,n]}$ where $BK_{i,1},\dots,BK_{i,n}$ are encryptions of the $n = \log_2(\ell)+3$ most
    significant bits of $\texttt{Phase}(h_i)$.
Output: A GLWE-GSW ciphertext $\mathbf{c}' \in H^\ell$ encoding the same bit as $c$ with polynomial noise.
 1: $\mathbf{x} \leftarrow f_{\mathbf{h}}^{-1}(c) \in \{0,1\}^\ell$
 2: $p \leftarrow 0$
 3: Set the initial state $(X_{0,0},...,X_{0,8\ell-1})$ where $X_{i,j} = 1$ iff $j \in [2\ell, 6\ell]$
 4: for each $i \in [1,\ell]$ s.t. $x_i = 1$ do
 5:     for $j = 1$ to $n$ do                          ▷ This loop adds $\texttt{Phase}(h_i)$ to the state
 6:         $p \leftarrow p+1$
 7:         for $k = 0$ to $8\ell - 1$ do          ▷ This loop adds $2^{n-j}$ to the state iff $BK_{i,j} = 1$
 8:             $X_{p,k} \leftarrow \texttt{GSW.Mux}(BK_{i,j}, X_{p-1,k-2^{n-j} \bmod 8\ell}, X_{p-1,k})$
 9:         end for
10:     end for
11: end for
12: return $\mathbf{c}' = X_{p,0}$                                  ▷ This is the final rounding.
```

permutation matrices. They had to encode the matrix product as small disjunctive normal forms, which increases the noise by the square root of the matrix dimension. In their algorithm, this dimension remains logarithmic due to their CRT decomposition, yet, it would have been polynomial if they had chosen the single cycle approach, and it would be much worse in the case of a deterministic automata, compared to our Lemma 7.11.

- Finally, our circuit represents a mirrored deterministic automata, and therefore, it has a single final state, which means that we get rounding for free. In [3], there are multiple accepting states, and summing them induces another polynomial noise factor in the final ciphertext.

Lemma 7.12 (Simple Bootstrapping) *Given a GLWE ciphertext $c \in H$, the gadget $\mathbf{h}$ and its function $f_{\mathbf{h}}^{-1}$, and the bootstrapping keys $(BK_{i,j})_{i\in[1,\ell],j\in[1,n]}$ where $BK_{i,1},\dots,BK_{i,n}$ are encryptions of the $n = \log_2(\ell)+3$ most significant bits of $\texttt{Phase}(h_i)$ with all $\|\texttt{Noise}(BK_{i,j})\|_\infty \leq B$, Alg. 6 outputs a GLWE-GSW ciphertext $\mathbf{c}' = \mathbf{z}' + \mu\mathbf{h} \in H^\ell$ encoding the same bit as c with noise $\|\texttt{Phase}(\mathbf{z}')\|_\infty \leq 2\ell^2(\log_2(\ell)+3)B$.*

Proof. Suppose $BK_{i,j} = Z_{i,j} + \mu_{i,j}\mathbf{h}$, and $X_{p,k} = Z'_{p,k} + \nu_{p,k}\mathbf{h}$ where all $Z_{i,j}$ and $Z'_{p,k}$ have small noises. By Lemma 7.5 and Line 8 in Alg. 6, we have $\left\|\texttt{Phase}(Z'_{p,k})\right\|_\infty \leq 2\ell \left\|\texttt{Phase}(Z_{i,j})\right\|_\infty + \mu_{i,j}\left\|\texttt{Phase}(Z'_{p-1,k-2^{n-j} \bmod 8\ell})\right\|_\infty + (1-\mu_{i,j})\left\|\texttt{Phase}((Z'_{p-1,k})\right\|_\infty$. By induction on p, since $\mu_{i,j} \in \{0,1\}$, it proves that each ciphertext in $X_p = (X_{p,0},...,X_{p,8\ell-1})$ has the same noise bound $\left\|\texttt{Phase}((Z'_{p,k})\right\|_\infty \leq 2\ell p B$. Alg. 6 ends with $p \leq \ell n = \ell(\log_2(\ell)+3)$. Thus, the noise of the output ciphertext is smaller than $2\ell^2(\log_2(\ell)+3)B$. □

Remember that under the hypothesis of lemma 7.6, the max-norm of the noise can be replaced by its square Gaussian parameter. This proves that the GLWE-GSW scheme is fully homomorphic according to Gentry's blueprint by design, as soon as the initial GLWE Gaussian parameter is $1/\tilde{O}(\ell^{1.5})$, which represents a time versus noise trade-off compared to [3] proposal, and has the advantage of giving the simplest possible decryption circuit as a proof of concept.

Overall, if we prefer to optimize the number of gates in the circuit, note that the function ϕ' which takes the $k = \tilde{O}(\log(\ell))$ bits of the CRT decomposition of an integer modulo $q \approx 8\ell$ and decides whether the encoded number is closer to $q/2$ than to 0 satisfies $\mathcal{N}(\phi') \leq qk$ for the exact same reason as in the whole decryption circuit. Thus, the following lemma gives an [3]-like variant of the decryption circuit with only $\tilde{O}(\ell)$ gates, and noise parameter $\tilde{O}(\ell^{1.5})$ instead of $\tilde{O}(\ell^2)$ in [3]. Note also that our CRT decomposition can be encoded in base 2, instead of being unary-encoded as in [3], which also saves a few logarithmic noise factors. More importantly, it expresses the whole bootstrapping as the composition of simple boolean functions, and obtains a time versus noise trade-off. The proof is still a consequence of Lemma 7.10, which only depends on simple intrinsic parameters of the underlying functions, i.e. the number of variables and the number of distinct partial functions.

Lemma 7.13 (CRT variant) *Given a GLWE ciphertext $c \in H$, the gadget $\mathbf{h}$ and its 1-bounded function $f_{\mathbf{h}}^{-1}$, Let $q = \prod_{i=1}^{t} p_i$ be an integer larger than 8ℓ where p_i are $t = O(\log(\ell))$ distinct primes where $p_i = O(\log(\ell))$. We suppose that the encryption of each individual bits $BK_{i,j,k}$ of $\lfloor q\mathtt{Phase}(h_i)\rceil \bmod p_j$ for $i \in [1, \ell]$ and $j \in [1, t]$, $k \in [0, \log_2(p_j)]$ are provided as bootstrapping key with $\|\mathtt{Noise}(BK_{i,j,k})\|_\infty \leq B$. Then given as input a ciphertext of a bit μ, one can compute a ciphertext $\mathbf{c} = \mathbf{z} + \mu\mathbf{h}$ of the same bit with noise $\|\mathtt{Phase}(\mathbf{z})\|_\infty = \tilde{O}(\ell^3)$ by evaluating $\tilde{O}(\ell)$ homomorphic Mux gates.*

Proof. Consider the following boolean functions:

- f_j for $j \in [1, t]$, takes ℓ numbers of $O(\log(p_j))$ bits and return the $O(\log(p_j))$ bits of their sum modulo p_j. (f_j can be viewed as $O(\log(p_j))$ boolean functions with a single bit output).
- ϕ' takes t numbers modulo $p_1, \ldots, p_t$, so $O(\log(\ell) \log\log(\ell))$ input bits, and returns 1 iff their CRT lift modulo q is closer to $q/2$ than to 0.

These functions correspond to simple arithmetic deterministic automata with respectively p_j and q states, whose current state after reading p bits is simply a linear combination with fixed coefficients of these bits modulo respectively p_j and q. Thus, the full BDD of f_j contains at most p_j nodes per level and the full BDD of ϕ' contains at most q nodes per level. Therefore, $\mathcal{N}(f_j) = O(\ell p_j \log(p_j)) = \tilde{O}(\ell)$ and $\mathcal{N}(\phi') = O(qt \log(p_j)) = \tilde{O}(\ell)$. To decrypt a GLWE ciphertext $c' = \sum_{k=1}^{\ell} x_i h_i \in H$, it suffices to homomorphically evaluate $y = \phi'(y_1, \ldots, y_t)$ where each $y_j = f(x_1.BK_{1,j}, \ldots, x_\ell.BK_{\ell,j})$. By lemma 7.10, the homomorphic ciphertext of each bit of y_j has noise norm $\tilde{O}(\ell^2)$, and thus, the output noise norm of y is $\tilde{O}(\ell^3)$. The total number of homomorphic Mux gates is $\sum_{j=1}^{t} \log(p_j)\mathcal{N}(f_j) + \mathcal{N}(\phi') = \tilde{O}(\ell)$ □

We also give comparisons of our FHE scheme to previous ones in Table 1. In that table, the group in GLWE is taken as $\mathbb{Z}_q^n$, which makes our scheme base on standard LWE assumption. In this case, we could take $\ell = O(n \log q)$.

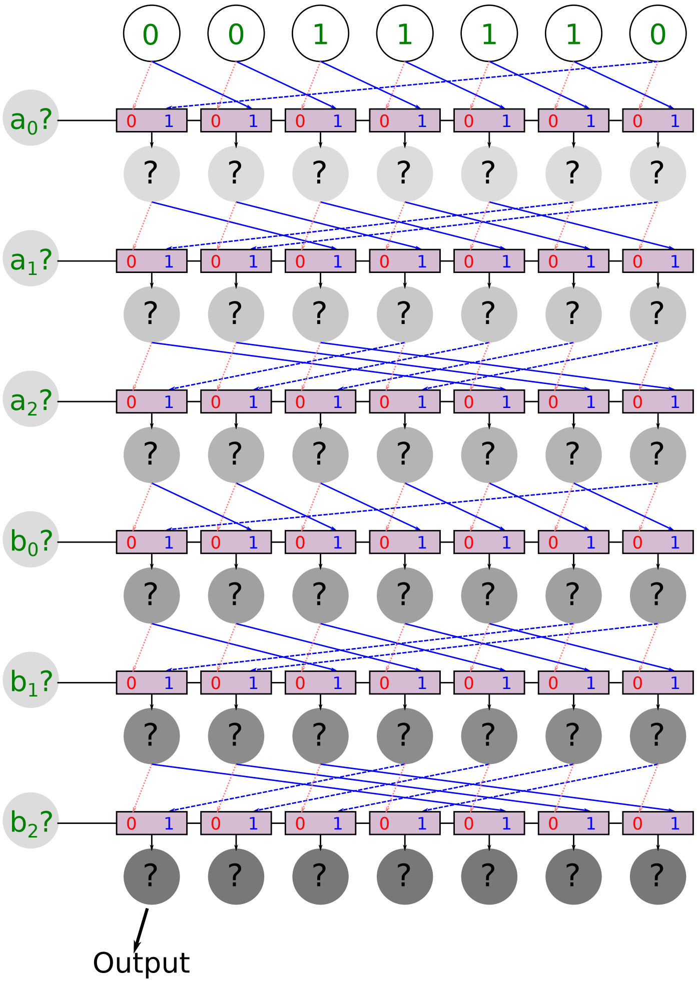
References

[1] M. Ajtai. Generating hard instances of lattice problems. In *STOC*, pages 99–108, 1996.

[2] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. of 29th STOC*, pages 284–293. ACM, 1997. Available at [16] as TR96-065.

[3] J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In *CRYPTO*, pages 297–314, 2014.

Figure 4: Schematics of the Mux-based decryption circuit



This circuit homomorphically sums two bitwise encrypted numbers $a = \underline{a_0a_1a_2}_2$ and $b = \underline{b_0b_1b_2}_2$ of $n = 3$ bits modulo $\ell = 7$, and returns 1 iff $(a + b)/\ell \mod 1$ is closer to $1/2$ than to 0. At each step, we choose to rotate or to preserve the state depending on the input encrypted bit value. The noise of each ciphertext, represented as gray levels, increases only linearly at each step, and remains polynomial in the last step.

Schemes	Primitive Gates	#Gates in Boots.	Boots. noise overhead
BGV12 [11]	And, Xor, Const.	$\tilde{O}(n^2)$	$n^{O(\log n)}$
Bra12 [10]	And, Xor, Const.	$\tilde{O}(n^2)$	$n^{O(\log n)}$
GSW13 [21]	And, Xor, Nand, Const.	$\tilde{O}(n^2)$	$n^{O(\log n)}$
BV14 [13]	And, Xor, Const.	$\tilde{O}(n^{6/\varepsilon})$	$\tilde{O}(n^{\varepsilon})$
AP14 [3]	And, Not, Const.	$\tilde{O}(n)$	$\tilde{O}(n^2)$
DM15 [15]	Nand, Const.	$\tilde{O}(n)$	$\tilde{O}(n^{1.5})$
Ours	Mux, Not, Const.	$\tilde{O}(n^2)$	$\tilde{O}(n)$
Ours (with CRT)	Mux, Not, Const.	$\tilde{O}(n)$	$\tilde{O}(n^{1.5})$

Table 1: Comparisons of LWE-based FHE Schemes

This table compares the primitive gates considered in the corresponding schemes, the number of homomorphic operations used to bootstrap, and the approximation factor of the underlying lattice problem. Note that Const. means a constant gates (*i.e.* noiseless ciphertexts) and $\varepsilon > 0$ is an arbitrarily small real number. The bootstrapping noise overhead is the ratio between the noise parameter of the refreshed ciphertext and the (fresh) noise of the bootstrapping key. This ratio must be multiplied by $O(\sqrt{n})$ to allow the evaluation of one additional primitive gate in a fully homomorphic scheme, and gives the required initial LWE inverse-error rate. It has to be multiplied by an additional $O(n)$ to get worst-case SIVP approximation parameters, under the quantum worst-case to average case reduction. The hidden constants or poly-logarithmic factors in [15] are smaller than in [3], but the underlying hard problem in [15] relies on ideal lattices. The group in our scheme is taken as $\mathbb{Z}_q^n$.

[4] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Proc. of Crypto '09*, LNCS 5677, pages 595–618. IACR, Springer-Verlag, 2009.

[5] L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.

[6] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.

[7] G. Baumslag, N. Fazio, A. Nicolosi, V. Shpilrain, and W. E. Skeith. Generalized learning problems and applications to non-commutative cryptography. In *Proc. ProvSec '11*, volume 6980 of *Lecture Notes in Computer Science*, pages 324–339. Springer, 2011.

[8] A. Becker, N. Gama, and A. Joux. A sieve algorithm based on overlattices. *LMS J. Comput. Math.*, 17(A):49–70, 2014. See also Cryptology ePrint Archive, Report 2013/685.

[9] D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In *Proc. of Crypto '96*, LNCS. IACR, Springer-Verlag, 1996.

[10] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *CRYPTO*, pages 868–886, 2012.

[11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.

[12] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D.Stehlé. Classical hardness of learning with errors. In *Proc. of 45th STOC*, pages 575–584. ACM, 2013.

[13] Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, pages 1–12, 2014.

[14] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In *Proc. of Asiacrypt*, pages 1–20, 2011.

[15] L. Ducas and D. Miccaincio. FHEW: Bootstrapping homomorphic encryption in less than a second. In *EUROCRYPT*, pages ??–??, 2015.

[16] ECCC. `http://www.eccc.uni-trier.de/eccc/`. The Electronic Colloquium on Computational Complexity.

[17] N. Fazio, K. Iga, A. R. Nicolosi, L. Perret, and W. E. Skeith III. Hardness of learning problems over burnside groups of exponent 3. *Designs, Codes and Cryptography*, pages 1–12, 2013.

[18] N. Gama and P. Q. Nguyen. Predicting Lattice Reduction. In *Proc. of Eurocrypt*, 2008.

[19] C. Gentry and S. Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In *Advances in Cryptology - Proc. EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer, 2011.

[20] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

[21] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO*, pages 75–92, 2013.

[22] D. Goldstein and A. Mayer. On the equidistribution of Hecke points. *Forum Math.*, 15(2):165–189, 2003.

[23] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.

[24] A. Lubotzky. The expected number of random elements to generate a finite group. *J. Algebra*, 257(2):452–459, 2002.

[25] D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM J. Comput.*, 34(1):118–169, 2004.

[26] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Proc. EUROCRYPT '12*, LNCS. IACR, Springer-Verlag, 2012.

[27] D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *Proc. CRYPTO '13*, volume 8042 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2013.

[28] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *SIAM J. Comput.*, 2007.

[29] L. J. Mordell. On some arithmetical results in the geometry of numbers. *Compositio Mathematica*, 1:248–253, 1935.

[30] P. Q. Nguyen and I. E. Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *J. Cryptology*, 15(3):151–176, 2002.

[31] P. Q. Nguyen and I. E. Shparlinski. Counting co-cyclic lattices. Preprint, 2013.

[32] I. Pak. On probability of generating a finite group. Preprint, 1999.

[33] A. Paz and C.-P. Schnorr. Approximating integer lattices by lattices with cyclic factor groups. In *Proc. of ICALP*, pages 386–393, 1987.

[34] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. of STOC '09*, pages 333–342. ACM, 2009.

[35] C. Peikert. An efficient and parallel gaussian sampler for lattices. In *Proc. of Crypto '10*, LNCS 6223, pages 80–97. Spinger-Verlag, 2010.

[36] M. Pohst. A modification of the LLL reduction algorithm. *J. Symbolic Comput.*, 4(1):123–127, 1987.

[37] O. Regev. Lattices in computer science #12: Average-case hardness. Regev's webpage, 2004.

[38] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[39] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43, 2010.

A Missing Proofs of Sect. 2

In order to ease the proofs of indistinguishability between two distributions, we introduce an approximate equality which will also be used in App. F: for $\varepsilon \in]0, \frac{1}{2}]$ and $n, m \in \mathbb{N}$, the notation $a =_{\varepsilon,n,m} b$ means $b \cdot \frac{(1-\varepsilon)^n}{(1+\varepsilon)^m} \leq a \leq b \cdot \frac{(1+\varepsilon)^n}{(1-\varepsilon)^m}$, which implies $b =_{\varepsilon,m,n} a$ and also $a = b(1 + (m+n)\varepsilon + O(\varepsilon^2))$. If $a =_{\varepsilon,n,m} b$ and $c =_{\varepsilon,n',m'} d$, we then have $ac =_{\varepsilon,n+n',m+m'} bd$.

With this notation, the main property of the smoothing parameter is that for all lattice L, $\mathbf{c} \in \operatorname{span}(L)$, $\sigma \geq \eta_\varepsilon(L)$, we have that $\rho_{\mathbb{R}^n,\sigma}(\mathbf{c} + L) =_{\varepsilon,1,0} 1/\operatorname{vol}(L)$. Thus for all overlattice $\bar{L} \supseteq L$ and all $\mathbf{c} \in \bar{L}$, $\rho_{\bar{L}/L,\sigma}(\mathbf{c}) =_{\varepsilon,1,1} \frac{\operatorname{vol}(\bar{L})}{\operatorname{vol}(L)}$

A.1 Proof of Prop. 2.2

We know that for $\varepsilon = 1/10$, independent Gaussian samples ($\mathbf{y}_i \leftarrow_\varepsilon \mathcal{D}_{L,s_i}$) such that $\sqrt{2}\eta_\varepsilon(L) \leq s_i$ have probability $\leq 9/10$ to be in any fixed hyperplane (see [37, Lemma 14]). This can be adapted to any $\varepsilon > 0$, so that one can extract a full-rank family F of n vectors of norm $\|F\| \leq \sqrt{n/2\pi} \cdot \max s_i$.

Then the generalized LLL algorithm for linearly dependent vectors [36] $F \cup B$ returns a basis C of length $\|C^*\| \leq \sqrt{n/2\pi} \cdot \max s_i$ in polynomial time.

A.2 Proof of Lemma 2.4 on Discrete Convolution

We now prove the dot product convolution Lemma.

The proof relies on the following equality: For $\alpha, \sigma \in \mathbb{R}_{ge0}$, for $\gamma = \left(\frac{1}{\sigma^2} + \frac{u^2}{\alpha^2}\right)^{-1/2}$ and $\Gamma = \sqrt{\alpha^2 + \sigma^2 u^2}$

$$\frac{1}{\sigma^2}x^2 + \frac{1}{\alpha^2}(t - ux)^2 = \frac{1}{\gamma^2}\left(x - \frac{\gamma^2 tu}{\alpha^2}\right)^2 + \frac{1}{\Gamma^2}t^2$$

Let $C = \mathbf{z} + L$ be some coset of a n-dimensional lattice L, $\mathbf{u} \in \mathbb{R}^n$, $\alpha, \sigma \in \mathbb{R}_{ge0}$ and $\varepsilon \in (0, 1/2)$. Let $(\mathbf{e}_1, \ldots, \mathbf{e}_n)$ be an orthonormal basis of $\mathbb{R}^n$ such that $\mathbf{u} = u \cdot \mathbf{e}_n$. A vector $\mathbf{v} \in \mathbb{R}^n$ will be expressed as $\sum_{i=1}^n v_i \mathbf{e}_i$.

$$\begin{aligned}
\sum_{\mathbf{v} \in C} \mathcal{D}_{\mathbb{R}^n,\sigma}(\mathbf{v}) \mathcal{D}_{\mathbb{R},\alpha}(t - \langle \mathbf{u}, \mathbf{v} \rangle) &= \sum_{\mathbf{v} \in C} \mathcal{D}_{\mathbb{R},\sigma}(v_1) \ldots \mathcal{D}_{\mathbb{R},\sigma}(v_n) \mathcal{D}_{\mathbb{R},\alpha}(t - uv_n) \\
&= \sum_{\mathbf{v} \in C} \mathcal{D}_{\mathbb{R},\sigma}(v_1) \ldots \mathcal{D}_{\mathbb{R},\sigma}(v_{n-1}) \frac{1}{\sigma\alpha} \exp\left(-\pi\left(\frac{1}{\sigma^2}v_n^2 + \frac{1}{\alpha^2}(t - uv_n)^2\right)\right) \\
&= \frac{1}{\sigma^n \alpha} \sum_{\mathbf{v} \in C} \exp\left(-\pi\left(\frac{1}{\sigma}v_1^2 + \cdots + \frac{1}{\sigma}v_{n-1}^2 + \frac{1}{\gamma^2}\left(v_n - \frac{\gamma^2 u}{\alpha^2}t\right)^2 + \frac{1}{\Gamma^2}t^2\right)\right)
\end{aligned}$$

Let f be the affine function which maps $\sum_{i=1}^n v_i \mathbf{e}_i$ to $\frac{v_1}{\sigma}\mathbf{e}_1 + \cdots + \frac{v_{n-1}}{\sigma}\mathbf{e}_{n-1} + \frac{v_n - \gamma^2 ut/\alpha^2}{\gamma}\mathbf{e}_n$. Then,

$$\begin{aligned}\sum_{\mathbf{v}\in C} \mathcal{D}_{\mathbb{R}^n,\sigma}(\mathbf{v})\mathcal{D}_{\mathbb{R},\alpha}(t-\langle \mathbf{u},\mathbf{v}\rangle) &= \frac{1}{\sigma^n\alpha}\sum_{\mathbf{v}\in C}\exp\left(-\pi\left(\|f(\mathbf{v})\|^2+\frac{1}{\Gamma^2}t^2\right)\right)\\ &= \frac{1}{\sigma^n\alpha}\sum_{\mathbf{v}'\in f(C)}\exp\left(-\pi\left(\|\mathbf{v}\|^2+\frac{1}{\Gamma^2}t^2\right)\right)\\ &= \frac{\Gamma}{\sigma^n\alpha}\mathcal{D}_{\mathbb{R}^n,1}(f(C))\mathcal{D}_{\mathbb{R},\Gamma}(t)\end{aligned}$$

Note that the largest eigenvalue of the linear part of f is $1/\gamma$, thus since $C=\mathbf{z}+L$, $f(C)=\mathbf{z'}+L'$ where $\eta_\varepsilon(L')\leq\eta_\varepsilon(L)/\gamma\leq 1$. Therefore, $\mathcal{D}_{\mathbb{R}^n,1}(f(C)) =_{\varepsilon,1,1} 1/\operatorname{vol}(L')=\sigma^{n-1}\gamma/\operatorname{vol}(L)$. We finally obtain:

$$\sum_{\mathbf{v}\in C}\mathcal{D}_{\mathbb{R}^n,\sigma}(\mathbf{v})\mathcal{D}_{\mathbb{R},\alpha}(t-\langle\mathbf{u},\mathbf{v}\rangle) =_{\varepsilon,1,1} \mathcal{D}_{\mathbb{R},\Gamma}(t)$$

Now, we can prove the lemma. First, let $\mathbb{K}=\mathbb{R}$ and $b\leftarrow\mathcal{D}_{\mathbb{K},\alpha,c+\langle\mathbf{u},\mathbf{v}\rangle}$ where $\mathbf{v}\leftarrow\mathcal{D}_{C,\sigma}$. Then the density of b is $\sum_{v\in C}\mathcal{D}_{\mathbb{K},\alpha}(b-c-\langle\mathbf{u},\mathbf{v}\rangle)\mathcal{D}_{C,\sigma}(\mathbf{v}) =_{\varepsilon,1,1} \mathcal{D}_{\mathbb{K},\Gamma}(b-c)=\mathcal{D}_{\mathbb{K},\Gamma,c}(b)$.

Second, let $\mathbb{K}=\frac{1}{N}\mathbb{Z}$, but assume that $\alpha\geq\eta_\varepsilon(\mathbb{K})$. the density of $b\in\mathbb{K}$ is

$$\sum_{\mathbf{v}\in C}\frac{\mathcal{D}_{\mathbb{R},\alpha}(b-c-\langle\mathbf{u},\mathbf{v}\rangle)}{\mathcal{D}_{\mathbb{R},\alpha}(\mathbb{K}-c-\langle\mathbf{u},\mathbf{v}\rangle)}\mathcal{D}_{C,\sigma}(\mathbf{v})$$

Since the denominator is $=_{\varepsilon,1,0} 1/\operatorname{vol}(\mathbb{K})$, the whole expression is nearly equal to $N\mathcal{D}_{\mathbb{R},\Gamma,c}(b)$. Since $\Gamma\geq\alpha\geq\eta_\varepsilon(\mathbb{K})$, then the N can be viewed as $1/\mathcal{D}_{\mathbb{R},\Gamma,c}(\mathbb{K})$, and finally, the density of b is $\mathcal{D}_{\mathbb{K},\Gamma,c}(b)$.

B Addendum on Sect. 3

B.1 Proof of Th. 3.1

Let $L\in\mathcal{L}_{G,m}$. Then G has rank $\leq m$ because $L\subseteq\mathbb{Z}^m$. And G is isomorphic to some product $\mathbb{Z}_{q_1}\times\cdots\times\mathbb{Z}_{q_k}$ of cyclic groups, where $1\leq k\leq m$, $q_i\geq 1$ and q_{i+1} divides q_i for all i. By [33], there exist primitive vectors $\mathbf{z}_1,\ldots,\mathbf{z}_k\in\mathbb{Z}^m$ s.t. $L=\{\mathbf{y}\in\mathbb{Z}^m,\ \langle\mathbf{y},\mathbf{z}_i\rangle\equiv 0 \pmod{q_i}, i\in[1,k]\}$. This shows that there exists $\mathbf{g}=(g_1,\ldots,g_m)\in G^m$ generating G such that $L=\mathcal{L}_\mathbf{g}$, where we recall that $\mathcal{L}_\mathbf{g}=\{(x_1,\ldots,x_m)\in\mathbb{Z}^m \text{ s.t. } \sum_{i=1}^m x_ig_i=0 \text{ in } G\}$.

Reciprocally, let $\mathbf{g}=(g_1,\ldots,g_m)\in G^m$ generate G. Consider the morphism ψ which maps $(x_1,\ldots,x_m)\in\mathbb{Z}^m$ to $\sum_{i=1}^m x_ig_i\in G$. By definition, the image of ψ is G and its kernel is $\mathcal{L}_\mathbf{g}$, therefore $\mathbb{Z}^m/\mathcal{L}_\mathbf{g}\simeq G$.

B.2 Proof of Lemma 3.2

Let $\mathbf{g}=(g_1,\ldots,g_m)\in G^m$ be such that the g_i's generate G. Let $\mathbf{h}=(h_1,\ldots,h_m)\in G^m$.

Assume that $\mathcal{L}_\mathbf{g}=\mathcal{L}_\mathbf{h}$. Define a map $\psi: G\to G$ as follows: for any $g\in G$, there exists a decomposition $\mathbf{x}=(x_1,\ldots,x_m)\in\mathbb{Z}^m$ s.t. $g=\sum_{i=1}^m x_ig_i$, and we let $\psi(g)=\sum_{i=1}^m x_ih_i$. This map is well-defined because if there are two decompositions of g, *i.e.* $g=\sum_{i=1}^m x_ig_i=\sum_{i=1}^m y_ig_i$, then $\mathbf{x}-\mathbf{y}\in\mathcal{L}_\mathbf{g}=\mathcal{L}_\mathbf{h}$, thus $\sum_{i=1}^m x_ih_i=\sum_{i=1}^m y_ih_i$ and $\psi(g)$ has the same value. It can be checked that ψ is a morphism. Since $\mathbb{Z}^m/\mathcal{L}_{\mathbf{g}} \simeq \mathbb{Z}^m/\mathcal{L}_{\mathbf{h}}$, we know that the h_i's generate G, and therefore ψ is an automorphism of G.

Reciprocally, let ψ be an automorphism of G such that $h_i = \psi(g_i)$ for all $1 \leq i \leq m$. Then $\psi(\sum_{i=1}^m x_i g_i) = \sum_{i=1}^m x_i h_i$ for any $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{Z}^m$. It follows that $\sum_{i=1}^m x_i g_i = 0$ if and only if $\sum_{i=1}^m x_i h_i = 0$. Hence, $\mathcal{L}_{\mathbf{g}} = \mathcal{L}_{\mathbf{h}}$.

Finally, unicity follows from $\psi(\sum_{i=1}^m x_i g_i) = \sum_{i=1}^m x_i h_i$.

B.3 Proof of Lemma 3.4

We prove the decisional version here, and the search version is analogous. Given the Decisional-GLWE$_{G,m',\alpha}(\mathcal{S})$ oracle (m' to be specified later), we now construct an efficient algorithm B to distinguish m samples $(a_i, b_i)_{i\in[1,m]}$ either from $A_{G,\beta}(\hat{s})$ or uniform in $G \times \mathbb{T}$ for some unknown β ($\leq \alpha$) and secret $\hat{s}$ sampled from $\mathcal{S}$. Let Z be the set of integer multiples of $\frac{1}{m^2}\alpha^2$ between 0 and α^2. For each $z \in Z$, B does the following. B picks m uniform samples $(\tilde{a}_i, \tilde{b}_i)_{i\in[1,m]}$ from $G \times \mathbb{T}$ and receives m samples $(a_i, b_i)_{i\in[1,m]}$ from his challenger. B estimates the acceptance probability of the Decisional-GLWE$_{G,m',\alpha}(\mathcal{S})$ oracle on the following two inputs: The first input is of the form $(\tilde{a}_i, \tilde{b}'_i)_{i\in[1,m]}$, where $\tilde{b}'_i \leftarrow \mathcal{D}_{\mathbb{T},z,\tilde{b}_i}$ and the second input is of the form $(a_i, b'_i)_{i\in[1,m]}$, where $b'_i \leftarrow \mathcal{D}_{\mathbb{T},z,b_i}$. If in any of these polynomial attempts a non-negligible difference is observed between two acceptance probabilities, output "non-uniform"; otherwise, output "uniform".

Note that $(\tilde{a}_i, \tilde{b}'_i)_{i\in[1,m]}$ is uniformly random in $G \times \mathbb{T}$. If $(a_i, b_i)_{i\in[1,m]}$ is uniformly random in $G \times \mathbb{T}$, then the two acceptance probabilities are exactly the same. If $(a_i, b_i)_{i\in[1,m]}$ is distributed as $A_{G,\beta}(\hat{s})$, then by classical convolution, $(a_i, b'_i)_{i\in[1,m]}$ is distributed as $A_{G,\sqrt{\beta^2+z}}(\hat{s})$. Consider the smallest $z \in Z$ such that $z \geq \alpha^2 - \beta^2$. Clearly, $z \leq \alpha^2 - \beta^2 + \frac{1}{m^2}\alpha^2$. Then

$$\alpha \leq \sqrt{\beta^2 + z} \leq \sqrt{\alpha^2 + \frac{1}{m^2}\alpha^2} \leq (1 + \frac{1}{m^2})\alpha.$$

Therefore, the statistical distance between $\mathcal{D}_{\mathbb{T},\alpha,\hat{s}(a_i)}$ and $\mathcal{D}_{\mathbb{T},\sqrt{\beta^2+z},\hat{s}(a_i)}$ is at most $O(\frac{1}{m^2})$ for $i \in [1, m]$. The statistical distance of m samples from $A_{G,\sqrt{\beta^2+z}}(\hat{s})$ and m samples from $A_{G,\alpha}(\hat{s})$ is at most $O(\frac{1}{m})$. Hence, for our choice of z, and by Chernoff bound, non-negligible $(O(\frac{1}{m}))$ difference will be observed with probability $\geq \frac{1}{3}$, if $(a_i, b_i)_{i\in[1,m]}$ is distributed as $A_{G,\beta}(\hat{s})$. Notice that we can set $m' = m^3$ in the Decisional-GLWE oracle.

B.4 Discretization of GLWE

In this subsection, we discuss discrete versions of the GLWE problem, where the LWE samples (a, b) are not taken in $G \times \mathbb{T}$, but b is instead chosen from a discrete subset of $\mathbb{T}$.

The first option is to use the rounded Gaussian distributions, which is suitable for a floating point representation. By convention, the distance between two numbers $x, y \in \mathbb{T}$ is $\min_{z\in\mathbb{Z}}(|x - y + z|)$. Let $h_1, \ldots, h_p$ be p real numbers such that $0 \leq h_1 < \cdots < h_p < 1$. We denote by H the values $h_1, \ldots, h_p$ mod 1, which is a finite subset of $\mathbb{T}$. We define the Rounded Discrete GLWE distribution denoted by $Ar_{G,H,\alpha}(\hat{s})$ the distribution of tha pair (a, b) over $G \times H$ where a is uniformly random in G and b is sampled according to $\mathcal{D}_{\mathbb{T},\alpha,\hat{s}(a)}$ and rounded to the nearest value over H. For the decisional variant, the uniform distribution of b over $\mathbb{T}$ is replaced by the distribution over H where b is sampled uniformly at random over $\mathbb{T}$ and rounded to its nearest value in H. With this definition, it is clear that starting from continuous GLWE (or uniform) samples $(a, b) \in G \times \mathbb{T}$, it suffices to take $a' = a$ and round b to its nearest value $b' \in H$ to obtain a discrete and rounded sample (a', b'). We denote by (Search) Decisional-GLWE$_r(G, H, \alpha)(\mathcal{S})$ the corresponding problems. If an oracle solves Decisional-GLWE$_r(G, H, \alpha)(\mathcal{S})$ (resp. Search), it automatically solves the underlying continuous Decisional-GLWE$_{G,\alpha}(\mathcal{S})$ (resp. Search) instance (provided that the solution remains unique). Reciprocally, one can turn a discrete rounded GLWE sample into a continuous one by adding some Gaussian noise larger than the maximal distance between two consecutive values in H:

Lemma B.1 *Let $h_1, \dots, h_p$ be p real numbers such that $0 \leq h_1 < \cdots < h_p < 1$ and H their representatives in $\mathbb{T}$. By convention, we set $h_{p+1} = 1 + h_1$. For all parameter β such that $\beta \geq \sqrt{2} \max_{i \in [1,p]}(h_{i+1} - h_i)$. Then there is a reduction from Decisional-*GLWE$_r(G, H, \alpha)(\mathcal{S})$ *(resp. Search) to Decisional-*GLWE$_{G,\sqrt{\alpha^2+\beta^2}}(\mathcal{S})$ *(resp. Search) for any distribution $\mathcal{S}$ over $\hat{G}$.*

The second option is to discretize GLWE over a finite subgroup $\mathbb{T}' = \frac{1}{N}\mathbb{Z}/\mathbb{Z}$ of the torus using the discrete Gaussian distribution. For $\beta > 0$ and some positive integer N, we denote $\bar{A}_{G,\beta,N}(\hat{s})$ the distribution over $G \times \mathbb{T}'$ which chooses $a \leftarrow G$ uniformly at random, sets $b \leftarrow \mathcal{D}_{\mathbb{T}',\beta,\hat{s}(a)}$ and outputs (a, b). We call (Search) Decisional-DGLWE$_{G,\alpha,N}(\mathcal{S})$ this discretization.

Again, we show that the discrete version is at least as hard as the continuous version for some suitable parameters:

Lemma B.2 *Let G be any finite abelian group and $N > 0$ an integer. Let $0 < \alpha, \beta < 1$ be reals such that $\beta \geq \eta_\varepsilon(\frac{1}{N}\mathbb{Z})$ for some negligible function ε. Then there is a reduction from Decisional-*GLWE$_{G,\alpha}(\mathcal{S})$ *(resp. Search) to Decisional-*DGLWE$_{G,\sqrt{\alpha^2+\beta^2},N}(\mathcal{S})$ *(resp. Search) for any distribution $\mathcal{S}$ over $\hat{G}$.*

Proof. The reduction does the following: given a sample $(a, b) \in G \times \mathbb{T}$, it sets $a' = a$ and samples $b' \leftarrow D_{\mathbb{T}',\beta,b}$. If the distribution of (a, b) is $A_{G,\alpha}(\hat{s})$, then $b \leftarrow D_{\mathbb{T},\alpha,\hat{s}(a)}$. Since $\beta \geq \eta_\varepsilon(\frac{1}{N}\mathbb{Z})$, the distribution of b' is statistically close to $\mathcal{D}_{\mathbb{T}',\sqrt{\alpha^2+\beta^2},\hat{s}(a)}$ by simple convolution. If (a, b) is uniformly random over $G \times \mathbb{T}$, then b is uniformly random over $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ and independent of a. We obtain that b' is uniformly random over $\mathbb{T}'$. □

C Missing Proofs of Sect. 4

C.1 Proof of Th. 4.2

Let B be an LLL-reduced basis with factor ε_{LLL} of an n-dimensional lattice $L \subseteq \mathbb{R}^n$. Let $\alpha = (1 + \varepsilon_{\text{LLL}})\sqrt{4/3}$.

Let $x_1, \dots, x_n$ denote the $\|\mathbf{b}_i^*\|$'s ordered by decreasing value. Let $k = \min \{i \in [1, n] \text{ st. } x_i/x_{i+1} > (1 + \varepsilon_{\text{LLL}})\sqrt{4/3}\}$, where $x_{n+1} = 0^+$.

Th. 4.2 follows from the following inequalities:

$$\text{bl}(L) \geq \left(\prod_{i=1}^{k} x_i\right)^{1/k}. \tag{11}$$

$$\delta_{\text{bl}(L)}(B) \leq \alpha^{\frac{n^2}{8}+\frac{n^2}{4}} \tag{12}$$

$$\delta_{\alpha^r \text{bl}(L)}(B) \leq \alpha^{\frac{(n-2r)^2}{8}+\frac{(n-2r)^2}{4}} \tag{13}$$

Proof of (11) The proof follows from the following two facts: First, we have: $\text{bl}(L) \geq \left(\prod_{j=n+1-i}^{n} \left\|\mathbf{b}_j^*\right\|\right)^{1/i}$ for all $i \in [1, n]$. Indeed, consider the projection π_{n+1-i} over $(\mathbf{b}_1, \dots, \mathbf{b}_{n-i})^\perp$. Then $\text{bl}(L) \geq \text{bl}(\pi_{n+1-i}(L))$ because projections cannot increase Gram-Schmidt norms, and $\text{bl}(\pi_{n+1-i}(L)) \geq \text{vol}(\pi_{n+1-i}(L)) = \left(\prod_{j=n+1-i}^{n} \left\|\mathbf{b}_j^*\right\|\right)^{1/i}$ because $B_{[n+1-i,n]}$

is a basis of $\pi_{n+1-i}(L)$. Second, let $A = \{i \in [1,n] \text{ s.t } \|\mathbf{b}_i^*\| \geq x_k\}$. By definition of k, $i \notin A \implies \|\mathbf{b}_i^*\| < x_k/((1+\varepsilon_{\text{LLL}})\sqrt{4/3})$. Therefore, Lovász' condition implies that for all $i \in A$, $i+1 \in A$. Thus, A is necessarily the right-most integer interval with k elements, *i.e.* $[n+1-k, n]$ and $\prod_{i=1}^{k} x_i = \prod_{i=n+1-k}^{n} \|\mathbf{b}_i^*\|$. □

Proof of (12) Let $\sigma_0 = \left(\prod_{i=1}^{k} x_i\right)^{1/k}$ and $j = \max\{i \in [1,k],\ x_i \geq \sigma_0\}$. Note that $\delta_{\sigma_0}(B) = \prod_{l=1}^{j} \frac{x_l}{\sigma_0} = \prod_{l=j+1}^{k} \frac{\sigma_0}{x_l}$. If $j \leq k/2$, then $x_l/\sigma_0 \leq \alpha^{j+1-l}$ for all $l \leq j$, therefore $\delta_{\sigma_0}(B) \leq \alpha^{(1+\ldots+j)} \leq \alpha^{\frac{k}{4}(\frac{k}{2}+1)}$. If $j > k/2$, $\sigma/x_l \leq \alpha^{l-j}$ for all $l \geq j$, therefore $\delta_{\sigma}(B) \leq \alpha^{(1+\ldots+(k-j))} \leq \alpha^{\frac{k}{4}(\frac{k}{2}+1)}$. In all cases, $\delta_{\sigma_0}(B) \leq \alpha^{\frac{k}{4}(\frac{k}{2}+1)} \leq \alpha^{\frac{n}{4}(\frac{n}{2}+1)}$. Finally, the cubicity-defect decreases with σ: since $\text{bl}(L) \geq \sigma_0$, $\delta_{\text{bl}(L)}(B) \leq \delta_{\sigma_0}(B)$. □

Proof of (13) Assume by contradiction that $\delta_{\alpha^r \text{bl}(L)} > \alpha^{(\frac{n}{2}-r)+\cdots+2+1}$. Let $j = \max\{i \text{ s.t. } x_i \geq \alpha^r \text{bl}(L)\}$, since $\delta_{\alpha^r \text{bl}(L)} \leq \frac{x_j}{\alpha^r \text{bl}(L)} \cdots \frac{x_1}{\alpha^r \text{bl}(L)} \leq \alpha\alpha^2 \ldots \alpha^j$ then $j > \frac{n}{2} - r$. Thus

$$\begin{aligned}\delta_{\text{bl}(L)}(B) &\geq \prod_{i=1}^{j} \frac{x_i}{\text{bl}(L)} \prod_{i=j+1}^{j+r} \frac{x_i}{\text{bl}(L)} \geq \delta_{\alpha^r \text{bl}(L)} \alpha^{rj} \alpha^{r-1} \ldots \alpha^1 \\ &> \alpha^{(\frac{n}{2}-r+r)+\cdots+(1+r)+(0+r)} \alpha^{(r-1)+\cdots+2+1} > \alpha^{\frac{n}{2}+\cdots+1}\end{aligned}$$

This contradicts (12), thus $\delta_{\alpha^r \text{bl}(L)} \leq \alpha^{\frac{(n-2r)^2}{8}+\frac{(n-2r)}{4}}$. □

C.2 Proof of Theorem 4.1 and Alg. 2

Let $a_i = \max(1, \|\mathbf{b}_i^*\|/\sigma)$ for $i \in [1,n]$. For each i from $k-1$ downto 1, we use the suffix "old" and "new" to denote respectively the values of the variables at the beginning and at the end of the "for" loop (line 3 of Alg. 2). Furthermore, we call x_i the value $\|\mathbf{c}_i^{*\text{new}}\|$ during iteration i. Note that x_i is also $\|\mathbf{c}_i^{*\text{old}}\|$ during the next iteration (of index $i-1$ since i goes backwards).

We show by induction over i that the following invariant holds at the end of each iteration (line 3 of Alg. 2):

$$a_i x_{i+1} \leq x_i \leq a_i x_{i+1} + \sigma a_i \tag{14}$$

At the first iteration ($i = k-1$), it is clear that $x_k = \|\mathbf{c}_k^{*\text{old}}\| = \sigma a_k$. At the beginning of iteration i (line 3), there are two cases:

1. if $\|\mathbf{c}_i^{*\text{old}}\| \leq \sigma$ and $\|\mathbf{c}_{i+1}^{*\text{old}}\| > \sigma$ (line 9), we size-reduce and swap the two vectors, so that $\|\mathbf{c}_i^{*\text{new}}\|$ satisfies:
$$\left\|\mathbf{c}_{i+1}^{*\text{old}}\right\| \leq \|\mathbf{c}_i^{*\text{new}}\| \leq \left\|\mathbf{c}_{i+1}^{*\text{old}}\right\| + \sigma.$$
Since $a_i = 1$, $x_{i+1} = \|\mathbf{c}_{i+1}^{*\text{old}}\|$ and $x_i = \|\mathbf{c}_i^{*\text{new}}\|$, the invariant (14) holds.

2. If $\|\mathbf{c}_i^{*\text{old}}\| > \sigma$ and $\|\mathbf{c}_{i+1}^{*\text{old}}\| > \sigma$ (line 7), we transform the block so that the norm of the first vector satisfies
$$R \leq \|\mathbf{c}_i^{*\text{new}}\| \leq R + \|\mathbf{c}_i^{*\text{old}}\|. \tag{15}$$
$$\text{where } R = \|\mathbf{c}_{i+1}^{*\text{old}}\| \, \|\mathbf{c}_i^{*\text{old}}\| / \sigma$$

   This condition can always be fulfilled with a primitive vector of the form $\mathbf{c}_i^{\text{new}} = \mathbf{c}_{i+1}^{\text{old}} + \alpha \mathbf{c}_i^{\text{old}}$ for some $\alpha \in \mathbb{Z}$. Since the volume is invariant, the new $\|\mathbf{c}_{i+1}^{*\text{new}}\|$ is upper-bounded by σ. And by construction, Equation (15) is equivalent to the invariant (14) since $\|\mathbf{c}_i^{*\text{old}}\| = a_i\sigma$, $\|\mathbf{c}_i^{*\text{new}}\| = x_i$ and $\|\mathbf{c}_{i+1}^{*\text{old}}\| = x_{i+1}$.

By expanding, this invariant implies that

$$x_1 \leq \sigma \sum_{i=1}^{k} a_1 \ldots a_i = \sigma \sum_{i=1}^{k} \delta_\sigma(B_{[1,i]}) \leq n\sigma\delta_\sigma(B)$$

Note that the transformation matrix of the unbalanced reduction algorithm is

$$\left[\begin{array}{cccc|ccc}
\alpha_1 & \cdots & \alpha_{k-1} & 1 & 0 & \cdots & 0 \\
1 & 0 & \cdots & 0 & \vdots & & \vdots \\
0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\
0 & 0 & 1 & 0 & 0 & \cdots & 0 \\
\hline
0 & \cdots & \cdots & 0 & 1 & 0 & 0 \\
\vdots & & & \vdots & 0 & \ddots & 0 \\
0 & \cdots & \cdots & 0 & 0 & 0 & 1
\end{array}\right]$$

where α_i is either $\lfloor -\mu_{i+1,i} \rceil$ or $\left\lceil -\mu_{i+1,i} + \frac{x_{i+1}}{\sigma}\sqrt{1-\frac{1}{a_i^2}} \right\rceil$. Since each x_{i+1} is bounded by $\delta_\sigma(B_{[i+1,n]})$, all coefficients have a size polynomial in the input basis and the overall complexity is therefore polynomial.

Now let us show (1). It suffices to prove that the following invariant holds at the beginning of each iteration:

$$\forall\ \nu \leq \sigma, \qquad \delta_\nu\left(C^{\text{old}}_{[i,n]}\right) \ \leq\ \delta_\nu\left(C^{\text{new}}_{[i,n]}\right) \ \leq\ \frac{\|\mathbf{c}_i^{*\text{new}}\|}{\sigma\delta_\sigma\left(C^{\text{old}}_{[i,n]}\right)} \times \delta_\nu\left(C^{\text{old}}_{[i,n]}\right) \tag{16}$$

Since all $\left\|\mathbf{c}_j^{*\text{old}}\right\| \leq \sigma$ for $j = i+2, \ldots, n$, $\delta_\sigma\left(C^*_{[i,n]}\right) = \delta_\sigma\left(C^*_{[i,i+1]}\right)$, where (*) is either old or new.

Hence, showing (16) amounts to show

$$\forall\ \nu \leq \sigma, \qquad \delta_\nu\left(C^{\text{old}}_{[i,i+1]}\right) \ \leq\ \delta_\nu\left(C^{\text{new}}_{[i,i+1]}\right) \ \leq\ \frac{\|\mathbf{c}_i^{*\text{new}}\|}{\sigma\delta_\sigma\left(C^{\text{old}}_{[i,i+1]}\right)} \times \delta_\nu\left(C^{\text{old}}_{[i,i+1]}\right) \tag{17}$$

Two cases can occur in the for loop of Alg. 2:

- First case: $\left\|\mathbf{c}_i^{*\text{old}}\right\| \leq \sigma$ and $\left\|\mathbf{c}_{i+1}^{*\text{old}}\right\| \geq \sigma$ (swap case). Since $\|\mathbf{c}_i^{*\text{new}}\|$ is projected on a space of higher dimension than $\left\|\mathbf{c}_{i+1}^{*\text{old}}\right\|$, we have $\|\mathbf{c}_i^{*\text{new}}\| \geq \left\|\mathbf{c}_{i+1}^{*\text{old}}\right\| \geq \nu$ and since the projected volume $\operatorname{vol}\left(C_{[i,i+1]}\right)$ remains unchanged, we have $\frac{\left\|\mathbf{c}_i^{*\text{new}}\right\|}{\left\|\mathbf{c}_{i+1}^{*\text{old}}\right\|} = \frac{\left\|\mathbf{c}_{i+1}^{*\text{old}}\right\|}{\left\|\mathbf{c}_{i+1}^{*\text{new}}\right\|}$ and hence $\left\|\mathbf{c}_{i+1}^{*\text{new}}\right\| \leq \left\|\mathbf{c}_i^{*\text{old}}\right\|$.

  By definition,

$$\begin{aligned}
\delta_\nu\left(C^{\text{new}}_{[i,i+1]}\right) &= \frac{\left\|\mathbf{c}_i^{*\text{new}}\right\|}{\nu} \times \max\left(1, \frac{\left\|\mathbf{c}_{i+1}^{*\text{new}}\right\|}{\nu}\right) \\
\delta_\nu\left(C^{\text{old}}_{[i,i+1]}\right) &= \frac{\left\|\mathbf{c}_{i+1}^{*\text{old}}\right\|}{\nu} \times \max\left(1, \frac{\left\|\mathbf{c}_i^{*\text{old}}\right\|}{\nu}\right)
\end{aligned}$$

We obtain $\frac{\delta_\nu\left(C^{\mathrm{new}}_{[i,i+1]}\right)}{\delta_\nu\left(C^{\mathrm{old}}_{[i,i+1]}\right)} = \frac{\left\|\mathbf{c}^{*\mathrm{new}}_i\right\|}{\left\|\mathbf{c}^{*\mathrm{old}}_{i+1}\right\|} \times \underbrace{\frac{\max\left(1, \frac{\left\|\mathbf{c}^{*\mathrm{new}}_{i+1}\right\|}{\nu}\right)}{\max\left(1, \frac{\left\|\mathbf{c}^{*\mathrm{old}}_{i}\right\|}{\nu}\right)}}_{\geq 1}$. Now $\sigma\delta_\sigma\left(C^{\mathrm{old}}_{[i,i+1]}\right) = \left\|\mathbf{c}^{*\mathrm{old}}_{i+1}\right\|$ proves the right side of (17).

For the left side, it suffices to notice that $\frac{\delta_\nu\left(C^{\mathrm{new}}_{[i,i+1]}\right)}{\delta_\nu\left(C^{\mathrm{old}}_{[i,i+1]}\right)} = \frac{\max\left(\left\|\mathbf{c}^{*\mathrm{new}}_i\right\|, \frac{\mathrm{vol}\left(C_{[i,i+1]}\right)}{\nu}\right)}{\max\left(\left\|\mathbf{c}^{*\mathrm{old}}_{i+1}\right\|, \frac{\mathrm{vol}\left(C_{[i,i+1]}\right)}{\nu}\right)} \leq 1.$

- Second case: $\left\|\mathbf{c}^{*\mathrm{old}}_i\right\| \geq \sigma$ and $\left\|\mathbf{c}^{*\mathrm{old}}_{i+1}\right\| \geq \sigma$. Thus we have $\|\mathbf{c}^{*\mathrm{new}}_i\| \geq \sigma$ and, $\left\|\mathbf{c}^{*\mathrm{new}}_{i+1}\right\| \leq \sigma$. Therefore, the two equality hold: $\delta_\sigma\left(C^{\mathrm{old}}_{[i,i+1]}\right) = \frac{\mathrm{vol}\left(C_{[i,i+1]}\right)}{\sigma^2}$, and a fortiori, $\delta_\nu\left(C^{\mathrm{old}}_{[i,i+1]}\right) = \frac{\mathrm{vol}\left(C_{[i,i+1]}\right)}{\nu^2}$. Since by definition, $\delta_\nu\left(C^{\mathrm{new}}_{[i,i+1]}\right) = \frac{\left\|\mathbf{c}^{*\mathrm{new}}_i\right\|}{\nu} \times \max\left(1, \frac{\left\|\mathbf{c}^{*\mathrm{new}}_{i+1}\right\|}{\nu}\right)$, the left side of (17) easily follows. Furthermore,

$$\begin{aligned}
\frac{\delta_\nu\left(C^{\mathrm{new}}_{[i,i+1]}\right)}{\delta_\nu\left(C^{\mathrm{old}}_{[i,i+1]}\right)} &= \frac{\nu\left\|\mathbf{c}^{*\mathrm{new}}_i\right\|}{\sigma^2\delta_\sigma\left(C^{\mathrm{old}}_{[i,i+1]}\right)} \times \max\left(1, \frac{\left\|\mathbf{c}^{*\mathrm{new}}_{i+1}\right\|}{\nu}\right) \\
&= \frac{\nu\left\|\mathbf{c}^{*\mathrm{new}}_i\right\|}{\sigma\delta_\sigma\left(C^{\mathrm{old}}_{[i,i+1]}\right)} \times \underbrace{\max\left(\frac{\nu}{\sigma}, \frac{\left\|\mathbf{c}^{*\mathrm{new}}_{i+1}\right\|}{\sigma}\right)}_{\leq 1}
\end{aligned}$$

This proves the right side of Inequality (17).

C.3 Proof of Th. 4.4

We will first prove the theorem using the first condition, which is tighter than the second one. The invariant of the main for loop is that at the beginning of i_{th} iteration, the current basis $C_{[i,n]}$ satisfies:

$$\left\|\mathbf{c}^*_j\right\| \leq q_j\sigma \text{ for all } j < i \qquad \text{and} \qquad \delta_\sigma(C_{[i,n]}) \leq \prod_{j=i}^{k} \frac{q_j}{n+1-j} \tag{18}$$

With this invariant, it is clear that the returned $\bar{B}$ at line 3 or 8 satisfies the upper-bound $\left\|\bar{B}^*\right\| \leq \sigma$.

Let us show (18) by induction on i. Clearly, the condition holds for $i = 1$.

At step 4, ℓ exists and is easy to compute, since the function $\nu \to \log(\delta_\nu(C_{[i,n]}))$ is a piecewise affine positive decreasing continuous function which is zero when $\nu = \left\|C^*_{[i,n]}\right\| \leq \|B^*\|$. With this ℓ, unbalanced reduction always produces a new basis such that $\|\mathbf{c}^{*\mathrm{new}}_i\| \leq q_i \cdot \sigma$. Then, there are two cases: either $\ell = \sigma$, and in this case, $\delta_\sigma(C^{\mathrm{new}}_{[i+1,n]}) = 1$. Or we have the equality $\ell\delta_\ell(C^{\mathrm{old}}_{[i,n]}) = q_i\sigma/(n+1-i)$. By replacing σ and ν by ℓ and σ respectively in (1), we obtain:

$$\delta_\sigma\left(C^{\mathrm{new}}_{[i,n]}\right) \leq \frac{\|\mathbf{c}^{*\mathrm{new}}_i\| \times \delta_\sigma\left(C^{\mathrm{old}}_{[i,n]}\right)}{\ell \times \delta_\ell\left(C^{\mathrm{old}}_{[i,n]}\right)} \leq \|\mathbf{c}^{*\mathrm{new}}_i\| \times \frac{\prod_{j=i}^{k} \frac{q_j}{n+1-j}}{\frac{q_i\sigma}{n-i+1}}$$

Since $\delta_\sigma\left(C^{\text{new}}_{[i,n]}\right) = \delta_\sigma\left(C^{\text{new}}_{[i+1,n]}\right) * \frac{\|b_i^{*\text{new}}\|}{\sigma}$, we have:

$$\delta_\sigma\left(C^{\text{new}}_{[i+1,n]}\right) \ \le \ \prod_{j=i+1}^{k} \frac{q_j}{n+1-j}$$

To prove the theorem with the second condition, it suffices to notice that at the first iteration, the length ℓ is smaller than $(n\,\|B^*\|^n/q_1)^{1/(n-1)}$, because for this value, $\ell\delta_\ell(B) \le \ell(\|B^*\|/\ell)^n = \frac{q_1}{n}$. Therefore, for the next iteration, $\left\|B^*_{[2,n]}\right\| \le \ell$ and $(\ell/\sigma)^{n-1} \le \frac{n\sigma}{q_1}\|B^*\|^n/\sigma^n \le q_2 \cdot \dots \cdot q_n/n^{k-1}$. The proof goes on by induction.

D Missing Proofs of Sect. 5

D.1 Proof of Th. 5.1

Calls to GSIS. Since $\sigma\eta_\varepsilon(\mathbb{Z}^n) \ge \text{bl}(L)\eta_\varepsilon(\mathbb{Z}^n) \ge \eta_\varepsilon(L)$, $\mathcal{D}_{\bar{L}/L,\sigma}$ is statistically close to the uniform distribution over $\bar{L}/L$. Therefore the $\mathbf{v}_i$'s are uniform $\mod L$ by Lemma 2.3. Thus, the elements $g_i = \varphi(\mathbf{v}_i)$ (defined at Line 5) have uniform distribution over G, which allows to make calls to the GSIS oracle at Line 6.
Correctness. It is easy to see that $\mathbf{v} = \sum_{i=1}^m x_i\mathbf{v}_i$ (defined in Line 7) is indeed a short vector of L, since $\varphi(\mathbf{v}) = \sum_{i=1}^m x_i g_i = 0$ and

$$E[\|\mathbf{v}\|] \le \|\mathbf{x}\| \times E[\|\mathbf{v}_i\|] \le \beta \times \sqrt{n/2\pi}\eta_\varepsilon(\mathbb{Z}^n)\sigma$$

Termination. It remains to prove that the algorithm indeed outputs n linearly independent vectors (and in particular that the output vectors are non-zero). This part of the proof is similar to [28]. The distribution of the output vectors $\mathbf{v}$'s depends on the $\mathbf{v}_i$'s and on the answer $\mathbf{x}$ of the GSIS-oracle, which only depends on $\mathbf{g} = (g_1, \dots, g_m)$. The distribution of $(\mathbf{v}_i), (g_i), \mathbf{x}$ during the algorithm can be equivalently simulated as follows: First choose $(g_1, \dots, g_m)$ uniformly in G, and call the GSIS oracle which returns a non-zero solution $\mathbf{x}$ with non-negligible probability. Now, for each g_i, sample the preimages $\mathbf{v}_i$, which necessarily have the conditional distribution of $\mathbf{v}_i \leftarrow D_{\bar{L},\sigma\eta_\varepsilon(\mathbb{Z}^n)}$ where $\varphi(\mathbf{v}_i) = g_i$, *i.e.* the distribution $D_{\varphi^{-1}(g_i),\sigma\eta_\varepsilon(\mathbb{Z}^n)}$ where $\varphi^{-1}(g_i)$ is a coset of L. From Proposition 2.2, since $\sigma\eta_\varepsilon(\mathbb{Z}^n) \ge \sqrt{2}\eta_\varepsilon(L)$, one can form a full rank family from $O(n)$ of such samples, which proves that the algorithm terminates.

D.2 Proof of Cor. 5.2

We consider two cases, depending on the rank k_n of G_n.

If $k_n \le n$ and $\#G_n \ge n^{k_n}\left(\eta_\varepsilon(\mathbb{Z}^n)\sqrt{2n/\pi}\beta_n\right)^n$, then it is a direct consequence of Th. 5.1.

Now, assume that $k_n > n$ and $\#G_n \ge n^{k_n}\left(\eta_\varepsilon(\mathbb{Z}^n)\sqrt{2n/\pi}\beta_n\right)^{k_n}$. Consider the decomposition of G_n into elementary divisors: $G_n \simeq \prod_{i=1}^{k_n}\mathbb{Z}_{q_i}$ where each q_{i+1} divides q_i. Then:

$$\left(\prod_{i=1}^{n} q_i\right)^{1/n} \ge \left(\prod_{i=1}^{k_n} q_i\right)^{1/k_n}.$$

Letting $H_n = \prod_{i=1}^n \mathbb{Z}_{q_i}$, we get that $\#H_n \ge \#G_n^{n/k_n} \ge n^n\left(\eta_\varepsilon(\mathbb{Z}^n)\sqrt{2n/\pi}\beta_n\right)^n$ and H_n has rank n. Therefore solving GSIS(H_n, m_n, β_n) with probability $\ge 1/\operatorname{poly}(n)$ can be used to solve worst-case n-dim ApproxSIVP$_{\eta_\varepsilon(\mathbb{Z}^n)\sqrt{n/\pi}\beta_n}$ But since $G_n \simeq H_n \times J_n$ for some finite abelian group J_n, we know that solving GSIS(H_n, m_n, β_n) with probability $\ge 1/\operatorname{poly}(n)$ can be reduced to solving GSIS(G_n, m_n, β_n) with probability $\ge 1/\operatorname{poly}(n)$.

E Direct reduction from BDD to Search-Group-LWE

By group switching (Sect. 6), there is a (quantum) reduction for Decisional-GLWE$_{G,\alpha}$ from worst-case GapSVP with approximation factor roughly $\tilde{O}(n^{1.5}/\alpha)$. This trivially gives us a quantum reduction for Search-GLWE$_{G,\alpha}$ from the same worst-case problem.

In this section, we show a direct classical reduction from BDD to Search-GLWE with slightly better approximation factors than the quantum one in Sect. 6.

Like GSIS, our hardness result for GLWE requires that the finite abelian group G is *explicit*. The main result of this section states that for appropriate choices of (G, m, α), if one can solve Search-GLWE$_{G,m,\leq\alpha}$ on average with probability $\geq 1/\operatorname{poly}(n)$, then one can quantumly approximate SIVP in the worst case, *i.e.* one can (quantum)-efficiently find short vectors in every n-dimensional lattice, which generalizes Regev's quantum Search-LWE reduction [38]. To do this, we only need to modify the classical part of Regev's proof, not the quantum part. More precisely, we only need to prove that a GLWE-oracle allows us to approximate bounded distance decoding (BDD) for dual lattices in the worst-case for some factor β: given a basis $B^\times$ of a dual lattice $L^\times$, and a target $\mathbf{t} \in \operatorname{span}(L^\times)$ within distance $\leq \beta\lambda_1(L^\times)$ to $L^\times$, find the lattice point $\mathbf{u} \in L^\times$ closest to $\mathbf{t}$.

Let us first explain the main difference with LWE. In previous proofs, the LWE-oracle is used to transform any β-BDD on $L^\times$ into an β/q-BDD over the same lattice $L^\times$. One iterates this process k times until the distance β/q^k becomes smaller than $2^{-O(n)}\lambda_1(L^\times)$, at which point Babai's nearest plane algorithm [5] solves the BDD instance in polynomial time. To allow arbitrary structures G, we reinterpret this as reducing β-BDD on $L^\times$ to β-BDD over $\bar{L}^\times$, where $\bar{L} = L/q$. Thus, instead of reducing the distance, we modify the lattice to increase $\lambda_1(L^\times)$ until the BDD instance can be solved by Babai's algorithm. This approach allows arbitrary overlattices $\bar{L}$, just like in our GSIS reduction.

More precisely, consider a BDD instance over $L^\times$: we have a target $\mathbf{t} \in \operatorname{span}(L^\times)$ close to some secret $\mathbf{u} \in L^\times$. Let $\hat{s} = \varphi^\times(\mathbf{u}) \in \hat{G}$. Remember that structural lattice reduction gives an exact sequence of groups $0 \to L \xrightarrow{\text{id}} \bar{L} \xrightarrow{\varphi} G \to 0$, where φ is efficiently computable. Let $\varphi^\times$ be as in Prop. 4.5. Like in the GSIS reduction, we sample short vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \bar{L}$ with Gaussian distribution, in such a way that each projection $g_i = \varphi(\mathbf{v}_i)$ is uniformly distributed over G. Then: $\hat{s}(g_i) = [\varphi^\times(\mathbf{u})](\varphi(\mathbf{v}_i)) \equiv \langle \mathbf{u}, \mathbf{v}_i\rangle \pmod 1$. Since $\mathbf{t}$ is close to $\mathbf{u}$, each $\langle \mathbf{t}, \mathbf{v}_i\rangle$ is therefore close to $\hat{s}(g_i) \bmod 1$, namely $\langle \mathbf{t}, \mathbf{v}_i\rangle - \hat{s}(g_i) \equiv \langle \mathbf{t} - \mathbf{u}, \mathbf{v}_i\rangle \pmod 1$. By adding a suitable noise, it is possible to simulate the distribution of the GLWE noisy approximation of $\hat{s}(g_i)$ (using Lemma 2.4). Then one can recover the character $\hat{s}$ by calling a Search-GLWE oracle: this allows to compute $\mathbf{u}' \in L^\times$ s.t. $\varphi^\times(\mathbf{u}') = \varphi^\times(\mathbf{u})$. One can compute $\mathbf{t} - \mathbf{u}'$, which is a target equally close to $\mathbf{u} - \mathbf{u}' \in \bar{L}^\times$, as $\mathbf{t}$ was close to $\mathbf{u} \in L^\times$. Hence, we have transformed a BDD-instance over $L^\times$ into a BDD-instance over $\bar{L}^\times$ with exactly the same error $\mathbf{t} - \mathbf{u}$. By iterating this process, one is eventually able to solve the BDD instance efficiently. Formally, we have:

Theorem E.1 *Let $n \in \mathbb{N}$, $\varepsilon = \operatorname{negl}(n)$, a BDD factor $\beta \leq \sqrt{\pi/2}\,(2n\eta_\varepsilon(\mathbb{Z}^n))^{-1}$ and $\theta = \beta\sqrt{2/\pi}\,(2n\eta_\varepsilon(\mathbb{Z}^n))$. Let $\alpha \in \left]\theta\sqrt{\pi/2}, \sqrt{\pi/2}\right]$ and an explicit finite abelian group G of rank $k \leq n$. Given as input a basis B of an n-dimensional lattice L and $\mathbf{t} \in \operatorname{span}(L^\times)$ such that the BDD instance $(B^\times, \mathbf{t})$ admits a unique solution $\mathbf{t} - \mathbf{w} \in L^\times$ with $\|\mathbf{w}\| \leq \beta\lambda_1(L^\times)$, and a Search-*GLWE$_{G,m,\leq\alpha}$ *oracle satisfying*

$$\#G \geq n^k \left(\frac{\|B^*\|}{\operatorname{bl}(L)} \frac{\theta\sqrt{\pi/2}}{\sqrt{2}\alpha}\right)^n, \tag{19}$$

Alg. 7 finds in time polynomial in n and $\log(1/\varepsilon)$ *a basis $\bar{B}$ of some overlattice $\bar{L}$ such that $\bar{L}/L \simeq G$, and a target $\bar{\mathbf{t}} \in \operatorname{span}(\bar{L}^\times)$ such that the BDD instance $(\bar{B}^\times, \bar{\mathbf{t}})$ have the same* *error* $\mathbf{w}$*, namely* $\bar{\mathbf{t}} - \mathbf{w} \in \bar{L}^\times$*. If* B *is LLL-reduced with factor* $\varepsilon_{_{LLL}}$*, we can replace* $\frac{\|B^*\|}{\mathrm{bl}(L)}$ *by* $((1+\varepsilon_{_{LLL}})\sqrt{4/3})^{\frac{n-1}{2}}$ *in* (19).

The proof of Th. E.1 is essentially summarized by Alg. 7, which makes a few simplifying assumptions.

Algorithm 7 Reducing BDD to GLWE

```
Input: A dimension $n$ and a negligible probability $\varepsilon = \mathrm{negl}(n)$, a basis $B$ of a $n$-dimensional integer lattice $L$, an
    average-oracle $\mathcal{O}$ for Search-GLWE$_{G,m,\le\alpha}$ satisfying the conditions of Th. E.1, a BDD factor $\beta$, a target $\mathbf{t}$ and an
    upper-bound $d_0 \le \beta\lambda_1(L^\times)$ on the error norm.
Output: a basis $\bar{B}$ of length $\|\bar{B}^*\| \le \|B^*\|/2$ of some ($G$-)overlattice $\bar{L}$ and a target $\bar{\mathbf{t}} \in \mathrm{span}(\bar{L}^\times)$ such that the BDD
    instance $(\bar{B}^\times, \bar{\mathbf{t}})$ has the same error $\mathbf{w}$ of norm $\le d_0$ than $(B^\times, \mathbf{t})$
1: $\sigma_0 \leftarrow \frac{\alpha}{\sqrt{2}d_0\eta_\varepsilon(\mathbb{Z}^n)} \ge \frac{\alpha\sqrt{2}}{\theta\sqrt{\pi/2}}\,\mathrm{bl}(L) \ge \sqrt{2}\,\mathrm{bl}(L)$.
2: Call structural lattice reduction (Alg.4) on $(B, G, \sigma_0)$ to get $(\bar{B}, \bar{L})$ and $\varphi : \bar{L} \to G$, $\varphi^\times : L^\times \to \hat{G}$ (Prop. 4.5)
3: repeat
4:     Sample $m$ random points $(\mathbf{v}_1, \cdots, \mathbf{v}_m) \in \bar{L}$ with distribution $\mathcal{D}_{\bar{L},\sigma_0\eta_\varepsilon(\mathbb{Z}^n)}$ using $\bar{B}$.
5:     Let $a_i = \varphi(\mathbf{v}_i)$ and $b_i \leftarrow \mathcal{D}_{\mathbb{T},\frac{\alpha}{\sqrt{2}},\langle\mathbf{t},\mathbf{v}_i\rangle}$, to form $(a_i, b_i)_{i\in[1,m]} \in (G \times \mathbb{T})^m$.
6:     Call the Search-GLWE$_{G,m,\le\alpha}$ oracle on $(a_i, b_i)_{i\in[1,m]}$ to find $\hat{s} \in \hat{G}$.
7: until Search-GLWE$_{G,m,\le\alpha}$ finds a solution
8: $\bar{\mathbf{t}} \leftarrow \mathbf{t} - \mathbf{u}$ where $\mathbf{u} \in \varphi^{\times-1}(\hat{s})$ (take any preimage modulo $\bar{L}^\times$)
9: return $\bar{B}, \bar{t}$
```

In Step. 6 of Alg. 7, the Search-GLWE$_{G,m,\alpha}$ oracle is called directly on the $(a_i, b_i)_{i\in[1,m]}$, whereas, strictly speaking, we should actually randomize these inputs to make sure that the solution $\mathbf{s}$ follows the right distribution: in the classical LWE reduction, one also uses the self-reducibility of LWE. To make sure that the input has the right distribution, the key step is Step. 5. Note that $\langle\mathbf{t}, \mathbf{v}_i\rangle = \langle\mathbf{u}, \mathbf{v}_i\rangle + \langle\mathbf{t} - \mathbf{u}, \mathbf{v}_i\rangle \bmod 1$, where the first term is equal to $\hat{s}(a_i) = \langle\varphi^\times(\mathbf{u}), \varphi(\mathbf{v}_i)\rangle$. Since $b_i \leftarrow \mathcal{D}_{\mathbb{T},\sqrt{\alpha}/2,\hat{s}(a_i)+\langle\mathbf{t}-\mathbf{u},\mathbf{v}_i\rangle}$ where $\mathbf{v}_i \leftarrow \mathcal{D}_{L,\sigma_0\eta_\varepsilon(\mathbb{Z}^n)}$, Lemma 2.4 proves that b_i has the requested distribution $\mathcal{D}_{\mathbb{T},\alpha',\hat{s}(a_i)}$ for some $\alpha' \le \alpha$

By iterating Alg. 7 and Th. E.1 a polynomial number of times, as the length of the input basis geometrically decreases, then $\lambda_1(L^\times)$ geometrically increases. Eventually, the BDD instance becomes easy, and the error $\mathbf{w}$ can be retrieved using for instance Babai nearest plane algorithm. Thus we deduce the following result on the hardness of Search-LWE.

Corollary E.2 *Let* $n \in \mathbb{N}$*,* $\varepsilon = \mathrm{negl}(n)$ *and two real sequences* $\beta_n \le \sqrt{\pi/2}\,(2n\eta_\varepsilon(\mathbb{Z}^n))^{-1}$*, and* $\alpha_n \in \left]\theta_n\sqrt{\pi/2}, \sqrt{\pi/2}\right]$ *where* $\theta_n = \beta_n\sqrt{2/\pi}\,(2n\eta_\varepsilon(\mathbb{Z}^n))$*. Let* $(G_n)_{n\in\mathbb{N}}$ *be a sequence of explicit finite abelian groups of rank* k_n*. If* $\#G_n \ge n^{k_n}\left((1+\varepsilon_{_{LLL}})\sqrt{4/3})^{\frac{n-1}{2}}\frac{\theta_n\sqrt{\pi/2}}{\sqrt{2}\alpha_n}\right)^{\max(n,k_n)}$*, then using polynomially many calls to an oracle solving Search-*GLWE$_{G_n,\le\alpha_n}$ *with probability* $1/\mathrm{poly}(n)$*, one can solve worst-case n-dimensional* BDD$_{\beta_n}$ *in (randomized) polynomial time and* ApproxSIVP$_{\sqrt{2}n/\beta_n}$ *in quantum polynomial time.*

F Missing proofs of Section E

F.1 Proof of Theorem E.1

Let $\mathbf{t}, B$ be the BDD-β instance on the dual $L(B)^\times$, and call $d_0 \le \beta\lambda_1(L^\times)$ an upper-bound on the error norm. Like in Theorem E.1, we suppose that $\beta \le \sqrt{\pi/2}\,(2n\eta_\varepsilon(\mathbb{Z}^n))^{-1}$, and call $\theta = \beta\sqrt{2/\pi}\,(2n\eta_\varepsilon(\mathbb{Z}^n)) < 1$.

In Alg. 7, the parameter $\alpha \in [\theta\sqrt{\pi/2}, \sqrt{\pi/2})$ is a valid noise parameter for GLWE oracles.

The parameter $\sigma_0 = \alpha/(\sqrt{2}d_0\eta_\varepsilon(\mathbb{Z}^n))$ is larger than $2n\beta/\sqrt{2}d_0$. Note that by Banaszczyk theorem [6], $\mathrm{bl}(L) \cdot \lambda_1(L^\times) \leq n$, so $\sigma_0 \geq \sqrt{2}\,\mathrm{bl}(L)$. Since $\#G$ is larger than $n^k(\|B^*\|/\sigma_0)^n$, one can indeed apply structural reduction to obtain a basis $\bar{B}$ of $\bar{L}$ such that $\|\bar{B}^*\| \leq \sigma_0$ (line 3 of Alg. 7).

There exists a (unique) vector $\mathbf{u} \in L^\times$ such that $\mathbf{t} = \mathbf{u} + \mathbf{w}$ with $\|\mathbf{w}\| \leq d_0$. We now prove that the instance $(a_i, b_i)_{i\in[1,m]}$ generated lines 6,7 is indistinguishable from a random $\mathrm{GLWE}(G, m, \leq \alpha)$ instance of solution $\hat{s} = \varphi^\times(\mathbf{u}) \in \hat{G}$. Namely the a_i's must be uniform in G, and for each $i \in [1, m]$, b_i must have distribution $\mathcal{D}_{\mathbb{T},\alpha,\hat{s}(a_i)}$.

The uniformity of the a_i's in G comes from the same reason as in Section 5, since they are isomorphic (by φ) to the $\mathbf{v}_i \mod L$, and the $\mathbf{v}_i$'s are drawn from a Gaussian distribution of parameter $\sigma_0\eta_\varepsilon(\mathbb{Z}^n) \geq \eta_\varepsilon(L)$. To show that the b_i's have the correct distribution, the idea is that $\hat{s}(a_i) = [\varphi^\times(\mathbf{u})](\varphi'(\mathbf{v}_i)) = \langle \mathbf{v}_i, \mathbf{u}\rangle \mod 1$. Suppose that a_i is fixed. Then the conditional distribution of $\mathbf{v}_i$ is $\mathcal{D}_{\varphi^{-1}(a_i),\sigma_0\eta_\varepsilon(\mathbb{Z}^n)}$ where $\varphi^{-1}(a_i)$ is a coset of L. Since the distribution of b is $\mathcal{D}_{\mathbb{T},\alpha/\sqrt{2},\langle\mathbf{t},\mathbf{v}_i\rangle}$ and $\langle \mathbf{t}, \mathbf{v}_i\rangle = \hat{s}(a_i) + \langle \mathbf{w}, \mathbf{v}_i\rangle$ where $\mathbf{v}_i$ has a discrete Gaussian distribution over a coset of L, then by the convolution Lemma 2.4, the distribution of b_i is at distance 4ε from the distribution $\mathcal{D}_{\mathbb{T},\nu,\hat{s}(a_i)}$ where the parameter ν is $= \sqrt{\alpha^2/2 + (\|\mathbf{w}\|\,\sigma_0\eta_\varepsilon(\mathbb{Z}^n))^2} \leq \alpha$.

Subsequent Iterations. Since the Search-GLWE oracle cannot distinguish the distribution of $(a_i, b_i)_{i\in[1,m]}$ from random GLWE samples, it will output the solution $\hat{s} = \varphi^\times(\mathbf{u})$ after a polynomial number of trials. Unfortunately, $\varphi^\times$ is not invertible, we can only recover $\mathbf{u}$ modulo $\ker(\varphi^\times) = \bar{L}^\times$. Let $\mathbf{u}_0$ be one preimage in ${\varphi^\times}^{-1}(\hat{s})$. The vector $\mathbf{t} - \mathbf{u}_0$ is now at distance $\leq d_0$ of $\bar{L}^\times$ instead of $L^\times$. Thus we can iterate the whole process by replacing L with $\bar{L}$.

Since $\mathrm{bl}(L)$ has decreased, the authorized interval for α increases, so α remains a valid noise parameter, and the same oracle may be used for all subsequent iterations.

Since the structural reduction always computes bases such that $\|B^*\|$ decreases by a constant factor compared to the previous basis, the while loop can be iterated $O(\log n)$ times, until $\|B^*\|$ becomes smaller than $1/d_0$. At this point, the BDD is very easy to solve exactly, for example using Babai nearest-plane algorithm.

F.2 Proof of Cor. E.2

Let $n \in \mathbb{N}$, and let $Q_n = (1 + \varepsilon_{\mathrm{LLL}})\sqrt{4/3})^{\frac{n-1}{2}} \frac{\theta_n\sqrt{\pi/2}}{\sqrt{2}\alpha_n}$. Like in Cor. 5.2, the case $k_n \leq n$ and $\#G_n \geq n^{k_n}(Q_n)^n$, is a direct consequence of (multiple iterations of) Th. E.1 and Regev's quantum connection between BDD_{β_n} and $\mathrm{ApproxSIVP}_{\sqrt{2}n/\beta_n}$.

Now, assume that $k_n > n$ and $\#G_n \geq n^{k_n}(Q_n)^{k_n}$. Again, from the decomposition of G_n into elementary divisors: $G_n \simeq \prod_{i=1}^{k_n} \mathbb{Z}_{q_i}$ where each q_{i+1} divides q_i, we can decompose G_n as $H_n \oplus J_n$ where the subgroup $H_n = \prod_{i=1}^{n} \mathbb{Z}_{q_i}$ has rank n and satisfies $\#H_n \geq n^n(Q_n)^n$. Note that any GLWE sample (a, b) on H_n with (unknown) secret $\hat{s}$ can be combined with a randomly generated GLWE sample (a', b') over J_n with a randomly chosen secret $\hat{s} \in \hat{J}_n$ to form a GLWE sample on G. Therefore solving $\text{Search-GLWE}(G, \alpha_n)$ with probability $\geq 1/\mathrm{poly}(n)$ can be used to solve $\text{Search-GLWE}(H_n, \alpha_n)$ with probability $\geq 1/\mathrm{poly}(n)$ which in turns can be used to solve worst-case n-dim BDD_{β_n}.

G Missing Proof of Sect. 6

G.1 Proof of Lemma 6.3

The main idea consists in the following: given an element of G, sample randomly an element of G' so that the evaluations on these two elements of the corresponding characters is almost preserved. The approximate equivalence of evaluations of characters comes from the duality of the maps φ' and $\varphi'^{\times}$. Given a sample $(a, b) \in G \times \mathbb{T}$, the procedure is as follows:

```
1: Choose one preimage $\mathbf{u} \in \varphi^{-1}(a)$ and sample $\mathbf{v} \leftarrow \mathcal{D}_{\bar{L}',r,\mathbf{u}}$ using the basis $\bar{B}'$ and Lemma 2.1. The preimage can be computed because $G$ is fully-explicit.
2: Let $a' = \varphi'(\mathbf{v})$.
3: Choose $b' \leftarrow \mathcal{D}_{\mathbb{T},rK,b}$.
4: Output $(a', b')$.
```

We now analyze the algorithm. We first show that the distribution of a' is nearly uniform in G'. It suffices to show that $\mathbf{v} \mod \mathbb{Z}^n \in \bar{L}'/\mathbb{Z}^n$ is (nearly) uniformly random. We note that, if $r \geq \eta_\varepsilon(\mathbb{Z}^n)$, we have that $\mathbf{v} \mod \mathbb{Z}^n$ is (almost) uniform (see [20]). However, this would require a very large r, which is not suitable for our reduction. Since a is uniform in G, a much smaller r is sufficient to show the uniformity of $\mathbf{v}$ mod $\mathbb{Z}^n$. Indeed, let $\mathcal{A} \in \bar{L}$ be a (finite) set containing exactly one representative of each class of $\bar{L}/\mathbb{Z}^n$. Note that $\varphi^{-1}(a)$ is a uniformly random coset $\mathbf{u} + \mathbb{Z}^n$ where $\mathbf{u} \in \mathcal{A}$. Let $\mathbf{v}_0 \in \bar{L}'$.

$$\begin{aligned}
\Pr[\mathbf{v} = \mathbf{v}_0 \text{ mod } \mathbb{Z}^n] &= \frac{1}{\#G} \sum_{a \in G} \mathcal{D}_{\bar{L}'/\mathbb{Z}^n,r,\varphi^{-1}(a)}(\mathbf{v}_0 + \mathbb{Z}^n) = \frac{1}{\#G} \sum_{\mathbf{u} \in \mathcal{A}} \mathcal{D}_{\bar{L}'/\mathbb{Z}^n,r,\mathbf{u}+\mathbb{Z}^n}(\mathbf{v}_0 + \mathbb{Z}^n) \\
&= \frac{1}{\#G} \sum_{\mathbf{u} \in \mathcal{A}} \sum_{\mathbf{z} \in \mathbb{Z}^n} \mathcal{D}_{\bar{L}',r,\mathbf{u}}(\mathbf{v}_0 + \mathbf{z}) = \frac{1}{\#G} \sum_{(\mathbf{u}-\mathbf{z}) \in \bar{L}} \mathcal{D}_{\bar{L}',r}(\mathbf{v}_0 + (\mathbf{z} - \mathbf{u})) \\
&= \frac{1}{\#G} \sum_{\mathbf{w} \in \bar{L}} \frac{\rho_{\mathbb{R}^n,r}(\mathbf{v}_0 + \mathbf{w})}{\rho_{\mathbb{R}^n,r}(\mathbf{v}_0 + \mathbf{w} + \bar{L}')} =_{2\varepsilon} \frac{1}{\#G} \cdot \frac{1/\operatorname{vol}(\bar{L})}{1/\operatorname{vol}(\bar{L}')} = \frac{1}{\#G} \cdot \frac{\#G}{\#G'} = \frac{1}{\#G'}.
\end{aligned}$$

Clearly, if the input b is uniformly random in $\mathbb{T}$, then b' is also uniform in $\mathbb{T}$. It remains to show that given as input a sample distributed from $A_{G,\alpha}(\hat{s})$, the algorithm outputs a sample distributed from $A_{G',\beta}(\hat{s}')$. Let $\mathbf{f} = \mathbf{v} - \varphi^{-1}(a)$, the distribution of $\mathbf{f}$ is $\mathcal{D}_{\bar{L}'-\varphi^{-1}(a),r}$. We also have, that $\hat{s}'(a') = [\varphi'^{\times}(\mathbf{s})](\varphi'(\mathbf{v})) = \langle \mathbf{s}, \mathbf{v} \rangle = \langle \mathbf{s}, \varphi^{-1}(a) \rangle + \langle \mathbf{s}, \mathbf{f} \rangle = \hat{s}(a) + \langle \mathbf{s}, \mathbf{f} \rangle$. Assume a' is fixed. Since b' is sampled from $\mathcal{D}_{\mathbb{T},rK,b}$ where $b \leftarrow \mathcal{D}_{\mathbb{T},\alpha,\hat{s}(a)}$, by classical convolution, the distribution of b' is $\mathcal{D}_{\mathbb{T},\sqrt{\alpha^2+(rK)^2},\hat{s}(a)}$ where $\hat{s}(a) = \hat{s}'(a') - \langle \mathbf{s}, \mathbf{f} \rangle$. Since $\mathbf{f}$ has Gaussian distribution over a coset, by the dot-product convolution lemma 2.4, the distribution of b' is statistically close to $\mathcal{D}_{\mathbb{T},\sqrt{\alpha^2+(\|\mathbf{s}\|r)^2+(rK)^2},\hat{s}'(a')}$

H Comparison with Previous Work of Ajtai and Micciancio

H.1 (G)SIS Reductions Based on Overlattices

Ajtai [1] and Micciancio [25] presented early reductions to (G)SIS, which are interesting to compare. Although their reductions are presented in a different manner, it turns out that they can both be recast in our simple overlattice framework:

1. Let $L \subseteq \mathbb{Z}^n$ be the "worst-case" lattice given by a basis B.

2. One constructs some overlattice $\bar{L}$ of L. This defines an exact sequence $0 \to L \xrightarrow{\text{id}} \bar{L} \xrightarrow{\varphi} G \to 0$, where φ is efficiently computable and G is some finite abelian group $\simeq \bar{L}/L$.

3. One has a sampling algorithm over $\bar{L}$ according to some distribution $\mathcal{D}$, such that its output $\mathbf{v} \in \bar{L}$ is short and $\varphi(\mathbf{v})$ has distribution statistically close to uniform over G.

4. One calls the sampler m times to obtain random short vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \bar{L}$. Then one calls the GSIS oracle on $(g_1, \ldots, g_m)$ where $g_i = \varphi(\mathbf{v}_i)$ has uniform distribution over G. This gives a short non-zero $\mathbf{x} \in \mathbb{Z}^m$ such that $\sum_{i=1}^m x_i g_i = 0$, which implies that $\mathbf{s} = \sum_{i=1}^m x_i \mathbf{v}_i \in L$. One analyzes the distribution of $\mathbf{s}$ to show that with non-negligible probability, $\mathbf{s}$ is non-zero, short (of norm significantly less than $\|B\|$) and does not lie in some hyperplane.

5. By iterating sufficiently many times this process, one eventually obtains a family of linearly independent short lattice vectors in L.

What differs among all such reductions is the choice of $\bar{L}$, sampler and distribution $\mathcal{D}$: in particular, the distribution dictates how short will be the solution vector $\mathbf{s}$.

H.2 Ajtai's SIS Reduction

We rewrite Ajtai's original reduction [1] in our framework.

Let L be the worst-case lattice given by a basis B. Ajtai constructs (see [1, Lemma 3]) a basis B' of a full-rank sublattice $L' \subseteq L$ such that B' is nearly-orthogonal, and $\|B'\|$ is not much bigger than $\|B\|$ (say, at most by some polynomial factor).

Here, the overlattice is $\bar{L} = q^{-1}L$ with $\bar{L}/L \simeq \mathbb{Z}_q^n = G$. But the sampling distribution $\mathcal{D}$ is not some discrete Gaussian distribution over $\bar{L}$ like in our GSIS reduction: instead, it is the uniform distribution over $\bar{L} \cap \mathcal{P}(B'/q)$, where $\mathcal{P}(B'/q)$ is the parallelepiped spanned by B'/q. Ajtai's sampler is very simple: pick random integers y_i in some large interval $[0, T]$ with uniform distribution, and reduce $\sum_{i=1}^n y_i \mathbf{b}_i / q \in \bar{L}$ inside the parallelepiped $\mathcal{P}(B'/q)$ using Babai's rounding algorithm, which implies that the sample has norm $\leq \sqrt{n}\|B'\|/q$. What is difficult is proving that the distribution obtained is statistically close to the uniform distribution over $\bar{L} \cap \mathcal{P}(B'/q)$, and that its image by φ is uniformly distributed over G.

H.3 Micciancio's GSIS Reduction

We rewrite Micciancio's reduction [25] in our framework: this is fairly different from the original description, and hopefully simpler.

Let $L \subseteq \mathbb{Z}^n$ be the worst-case lattice given by a basis B. Micciancio [25] considers a special lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ for which CVP can be solved in polynomial time, and uses this CVP subroutine to construct (see [25, Lemma 2.11]) a nearly-orthogonal basis M of a full-rank sublattice $\mathcal{L}'$ of $\mathcal{L}$: for instance, if $\mathcal{L} = \mathbb{Z}^n$, then $M = qI_n$. This defines a finite abelian group $G' = \mathcal{L}/\mathcal{L}'$. Similarly, using Babai's nearest plane algorithm instead of a CVP subroutine, Micciancio also constructs a nearly-orthogonal basis C of a full-rank sublattice L' of L.

Let ψ be the (non-singular) linear transformation over $\mathbb{R}^n$ mapping M to C, *i.e.* $\psi(\mathbf{x}) = \mathbf{x}M^{-1}C$, then $\psi(\mathcal{L}') = L'$. $\mathcal{L}'$ is a sublattice of both $\mathcal{L}$ and $\psi^{-1}(L)$, and similarly, L' is a sublattice of both L and $\psi(\mathcal{L})$. Notice that $\mathcal{L}/\mathcal{L}' \simeq \psi(\mathcal{L})/\psi(\mathcal{L}') = \psi(\mathcal{L})/L'$ so $\psi(\mathcal{L})/L' \simeq G'$.

Here, the overlattice is $\bar{L} = L - \psi(\mathcal{L})$ formed by all sums of the form $\mathbf{v} + \mathbf{w}$ where $\mathbf{v} \in L$ and $\mathbf{w} \in \psi(\mathcal{L})$: one can check that $\bar{L}$ is indeed an overlattice of L. By the second and third isomorphism theorems, we have:

$$\bar{L}/L \simeq \psi(\mathcal{L})/(\psi(\mathcal{L}) \cap L) \simeq (\psi(\mathcal{L})/L')/((\psi(\mathcal{L}) \cap L)/L') \simeq G'/G'',$$

for some subgroup $G'' \simeq (\psi(\mathcal{L}) \cap L)/L'$ of G'. So the target group is $G = G'/G''$. Note that a GSIS-oracle for G' implies a GSIS-oracle for G.

The sampler has support $\bar{L} \cap k(n)W$ where $W = \psi(V)$ and V is the (closed) Voronoi cell of $\mathcal{L}$ and $k(n)$ is some integer function growing logarithmically. In fact, the sampler is based on a "local" sampler given in [25]: the output $\mathbf{v}$ of the global sampler is formed by adding $k(n)$ vectors of the form $\mathbf{v}'_j - \psi(\mathbf{w}_j)$, where $(\mathbf{v}'_j, \mathbf{w}_j) \in \bar{L} \times \mathcal{L}$ is an output for the local sampler given in [25, Lemma 6.6]. This is because the local sampler [25, Lemma 6.6] only produces a nearly-uniform distribution, so one sums $k(n)$ vectors to guarantee a distribution sufficiently close to uniform for $\varphi(\mathbf{v})$.

The local sampler given by [25, Lemma 6.6] is a bit technical: compared to Ajtai's, the main idea is to use the CVP-solver of $\mathcal{L}$ to reduce the norm of the sample, which is why $\psi(V)$ matters. By analyzing carefully the distribution of the local sampler, Micciancio shows that the expectation of the norm of the final linear combination $\mathbf{s}$ is $o(\|B\|)$.

H.4 Comparison with Structural Reduction

Both Ajtai's reduction and Micciancio's reduction involve the construction of a good basis of some sublattice: we note that this is reminiscent of structural reduction. Indeed, recall that given a basis B of a lattice L and a (sufficiently large) finite abelian group G, structural reduction finds a basis $\bar{B}$ of some overlattice $\bar{L}$ such that $\bar{L}/L \simeq G$ and $\|\bar{B}^*\|$ is much smaller than $\|B^*\|$. By duality, structural reduction also allows to find a basis B' of some sublattice L' of L such that $L/L' \simeq G$ and $\min_i \|\mathbf{b}'^*_i\|$ is much larger than $\min_i \|\mathbf{b}^*_i\|$. In other words, finding a good basis of some sublattice with prescribed structure is equivalent to structural lattice reduction. However, the sublattice constructions of Ajtai [1, Lemma 3] and Micciancio [25, Lemma 2.11] do not allow to choose the structure of the sublattice, which is not as powerful as structural reduction.

Furthermore, although Ajtai's reduction and Micciancio's reduction both allow to sample short lattice vectors in some overlattice, it must be stressed that the target group G of the overlattice cannot be arbitrarily chosen. The restrictions for G in Ajtai's reduction are the same as in the GPV reduction [20]: $G = (\mathbb{Z}_q)^n$ or $G = \prod_{i=1}^n q_i$ where the q_i's are prime numbers of similar size. In Micciancio's reduction, G is implicitly defined by the CVP algorithm and the sublattice algorithm of [25, Lemma 2.11]: in principle, if we ignore technical issues regarding the distribution of $\varphi(\mathbf{v})$, we could select $\mathcal{L} = \mathbb{Z}^n$ to have more freedom over G, but we would need a procedure to find a good basis of some integer lattice $\mathcal{L}' \subseteq \mathbb{Z}^n$ such that $\mathbb{Z}^n/\mathcal{L}' \simeq G$, which is missing in [25]. In particular, for both Ajtai's reduction and Micciancio's reduction, the case of a cyclic group G of large prime order looks unreachable, unlike structural reduction.

I Relationships Between GSIS, GLWE and HNP

We clarify the relationships between GSIS, GLWE and the Hidden Number Problem (HNP).

I.1 From Decisional-GLWE to GSIS

There is a folklore reduction from Decisional-GLWE$_{G,\alpha}$ to GSIS(G, m, β) when $0 < \alpha\beta \cdot \sqrt{m} < 1$. Given samples $(a_i, b_i) \in G \times \mathbb{T}$ for $1 \leq i \leq m$ from Decisional-GLWE$_{G,\alpha}$, use the GSIS(G, m, β) oracle on $(a_1, ..., a_m)$ to find a non-zero vector $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{Z}^m$ such that $\sum_{i=1}^m x_i a_i = 0 \in G$ and $\|\mathbf{x}\| \leq \beta$. Now consider the summation $t \triangleq \sum_{i=1}^m x_i b_i \in \mathbb{T}$. If $b_i = \hat{s}(a_i) + e_i$ for some $\hat{s} \in \hat{G}$ and $e_i \leftarrow \mathcal{D}_{\mathbb{T},\alpha}$, then $t = \hat{s}(\sum_{i=1}^m x_i a_i) + \sum_{i=1}^m x_i e_i = \sum_{i=1}^m x_i e_i \in \mathbb{T}$. Since $\|\mathbf{x}\| \leq \beta$ and $\|\mathbf{e} = (e_1, ..., e_m)\| \leq \alpha \cdot \sqrt{m}$ (with overwhelming probability), we have that $|t| \leq \alpha\beta\sqrt{m}$. On the other hand, if the b_i's are uniform in $\mathbb{T}$, then t will also be uniform over $\mathbb{T}$. Therefore, the distinguisher for Decisional-GLWE$_{G,\alpha}$ simply looks at t and answers that the input (a_i, b_i)'s are a GLWE sample if the absolute value is at most $\alpha\beta\sqrt{m}$, and answers uniform otherwise. The distinguisher will succeed with advantage $\geq 1 - \alpha\beta\sqrt{m}$.

For simplicity, one can choose $\alpha\beta\sqrt{m} = 1/2$. Since the underlying worst-case n-dimensional GapSVP in the quantum reduction for Decisional-GLWE$_{G,\alpha}$ has approximation factor $\tilde{O}(n^{1.5}/\alpha)$, the analysis above gives us a *quantum* reduction from worst-case GapSVP with approximation factor $\tilde{O}(n^{1.5}\sqrt{m}\beta)$ to GSIS(G, m, β). On the other hand, the result in Sect. 5 shows a direct *classical* reduction from worst-case GapSVP with approximation factor $\tilde{O}(\sqrt{n}\beta)$ to GSIS(G, m, β), which is better.

I.2 Relations betweens HNP and LWE

Cor. 6.4 reduces GLWE from one group to another. Since GLWE over a cyclic prime-order group is exactly a randomized version of the Hidden Number Problem (HNP), Cor. 6.4 shows that this randomized version of HNP is equivalent to decisional-LWE (using both directions of Cor. 6.4). We now compare this general HNP hardness result to previous results of Brakerski *et al.* [12].

First, as we mentioned in the introduction, [12, Cor 3.4] established the hardness for the hidden number problem when the large prime p is replaced by q^n where q is smooth: the nice modulus-dimension switching technique of [12, Cor 3.4] allows to transfer LWE samples to HNP samples with a power modulus.

Though this is not explicitly mentioned in [12], it turns out that one can prove that decisional-LWE can be reduced to HNP by combining [12, Cor 3.3] and [12, Cor 3.4], albeit with worse bounds than our general result.

In the other direction, surprisingly, one can also establish a reduction from HNP to decisional-LWE by carefully combining several results of [12]. More precisely, taking $k = 1$ in [12, Th. 4.1] reduces HNP to LWE with binary secret and large modulus. Next, one could switch to a small modulus and use self-reduction to obtain normal LWE samples. However, this reduction is again indirect, and requires to work with a large intermediate group.

More generally, [12, Th. 4.1] is indirect, and Cor. 3.2, 3.3, 3.4 and Th. 4.1 of [12] are all particular cases of our Cor. 6.4: in fact, any combination of these statements can be achieved with a "single pass" of our Cor. 6.4, which therefore leads to a better output noise. One key point, compared to [12], is that for each n and each classical secret space of GLWE containing less than K^n elements (*e.g.*: Gaussian, interval, full space, *etc.*), there exists at least one morphism $\varphi^\times$ on $\mathbb{Z}^n$ so that the secret space is $\sqrt{n} \cdot K$ bounded (An explicit example is to take the morphisms corresponding to the decomposition of coordinates in base K.). Because of this, a single pass of our theorem is sufficient to prove the equivalence between HNP and LWE, without having to go through the long and technical proof of [12], without requiring a large intermediate group, and without having to care about arithmetic complications related to the modulus as in [12, Th. 4.1].