SoK: Learning with Errors, Circular Security, and Fully Homomorphic Encryption

Daniele Micciancio[1(✉)] and Vinod Vaikuntanathan[2]

[1] UC San Diego, La Jolla, CA, USA
daniele@cs.ucsd.edu
[2] MIT, Cambridge, MA, USA

Abstract. All known constructions of fully homomorphic encryption (FHE) schemes from the learning with errors (LWE) assumption require the encryption schemes to be circular secure. A long-standing open problem in the study of FHE schemes is to demonstrate evidence for their circular security. In this work, we systematize the flavors of circular security required for a number of FHE constructions, formulate circular security conjectures, show search-to-decision reductions for them, and pose several open problems.

1 Introduction

The celebrated notions of semantic security and indistinguishability of encryption schemes, first postulated by Goldwasser and Micali [33], assume that the message to be encrypted cannot depend on the private decryption key. Indeed, the dangers of encrypting messages that the adversary cannot herself come up with was already pointed out in their work [33, Section 5.1]. Nearly two decades later, Black, Rogaway and Shrimpton [8] initiated the formal study of security of encryption schemes with key-dependent messages, or KDM security, which requires that encryption schemes remain semantically secure—equivalently, IND-CPA secure—in the presence of ciphertexts that encrypt functions of the private decryption key. It is not hard to construct encryption schemes which are secure in the standard sense of indistinguishability, but completely insecure in the presence of encryptions of key-dependent messages.[1] So, KDM security is certainly a theoretically non-trivial notion. Moreover, while KDM security may seem at first an esoteric concern, it arises both naturally—in the context of full disk encryption where the private keys on disk may inadvertently get encrypted under themselves, and in the symbolic analysis of protocols [1,8,47]—and by design—in the context of certain anonymous credential systems [18].

[1] Indeed, consider a (private-key) encryption scheme where the encryption algorithm works as normal, except it checks if its input message is the private key, and if so, acts as the identity function outputting the private key. In the presence of the encryption of the private key, this scheme is clearly insecure. However, security is maintained in the absence of any such circular encryption.

Q. Tang and V. Teague (Eds.): PKC 2024, LNCS 14604, pp. 291–321, 2024.
https://doi.org/10.1007/978-3-031-57728-4_10

In 2009, circular security[2] found a new, prominent application: the construction of Fully Homomorphic Encryption (FHE) schemes, namely, encryption schemes that allow one to perform arbitrary computations on encrypted data. It is this latter application, namely circular security in the context of fully homomorphic encryption, that is the focus of this paper. While the functional and security requirements of FHE do not explicitly require circular security, Gentry's bootstrapping procedure (underlying his first candidate FHE proposal [29] and a long sequence of follow up works, e.g., see [3,10,12,16,17,20,23,25,31,36]) makes essential use of circular security. More specifically, an encryption system that can support computation of arbitrary polynomial-size circuits, for a fixed set of encryption parameters, is called a *fully homomorphic* encryption (FHE) scheme. In contrast, a weaker type of homomorphic encryption is called *leveled* homomorphic if for every depth parameter d (which is polynomial in a security parameter), there is a set of encryption parameters that support computation of depth-$\leq d$ circuits. The point is, in leveled homomorphic encryption schemes, the size of the encryption parameters grow with d, and can only support homomorphic evaluation of circuits of depth at most d.

The first candidate construction of an FHE scheme was proposed by Gentry [29]. Starting from the work of Brakerski and Vaikuntanathan [16], we have several *leveled HE* schemes [3,10,12,17,20,31,36] whose security is based on the hardness of the learning with errors (LWE) problem, even with a polynomial modulus [17]. However, to this date, rather frustratingly, the only way we know to make them fully homomorphic goes via Gentry's bootstrapping procedure which requires making public a circular encryption of the private key. Even the plain semantic security of one of these encryption schemes seems to require circular-type assumptions [36].

Embarrassingly, more than a decade later, we still do not know how to prove the circular security of any of these leveled HE schemes short of simply assuming it. Indeed, the only constructions of (pure, or non-leveled) FHE schemes we know, with the exception of a construction based on indistinguishability obfuscation (IO) (namely, [19], instantiated with the IO candidates of [39,40]), require assuming circular security. The situation has gotten steadily worse: while circular security assumptions for all the FHE schemes listed above have a similar flavor, the technical details are often different due to the different encryption schemes and/or the different encoding of the private key that each variant of each scheme demands. Even formulating the exact circular security assumption often requires first defining the homomorphic encryption scheme, and then expressing the assumption *in terms of it*. This makes the assumptions hard to understand and study, and has had downstream consequences. First, the standard that one expects with new hardness assumptions in cryptography is that they are followed with adequate cryptanalysis, including the description of challenge

[2] In this work, we will use the terms KDM security and circular security interchangeably, although the latter has been used in the literature to refer to encrypting some representation of the private key itself, whereas the former refers to more complex functions of the private key.

instances much like the RSA challenge [41] or the Darmstadt Lattice Crypto challenge [22]. Unfortunately, such "due diligence" has not been followed for the various circular security assumptions and indeed, a major bottleneck in doing so is the lack of even a systematic understanding of *what* these assumptions are. Secondly, versions of circular secure LWE were formulated in the context of building indistinguishability obfuscation [26,58], and claimed to be similar to FHE assumptions, but then broken shortly after [38]. This again is arguably due to a lack of systematic understanding of circular security assumptions.

In this paper, we take a *first step* to remedy this situation. The primary goal of this paper is to formulate (one or more) LWE circular security assumptions, just at the level of the LWE problem, and show that known FHE schemes can be proved secure based on such an assumption. This has the following advantages:

1. It provides a simple, concrete assumption (similar to LWE) that can be understood and investigated without having to fully describe an FHE scheme.
2. Such an effort potentially allows us to reduce the security of multiple FHE constructions to a single assumption (or a small set of assumptions), possibly in an efficient manner, relating the concrete security of several FHE schemes to the concrete security of the assumption.
3. If different assumptions are needed by different schemes, a systematic study lets us investigate possible reductions between assumptions, allowing to compare the strength of the assumption underlying different encryption schemes.
4. It offer a basis to generate concrete challenges, similar to Darmstadt lattice challenges (see `latticechallenge.org`).
5. It allows us to consider simpler, possibly weaker circular security assumptions, not necessarily enough to build FHE, but perhaps allowing a reduction to standard LWE or worst-case lattice problems.

Our concrete contributions are as follows.

1. We put forth a circular security conjecture called quadratic circular LWE or, succinctly, circLWE.
2. We show several properties of circLWE including: the proof of a weaker version, namely linear circular LWE, under the standard LWE assumption; a search to decision reduction; and a proof of security of a stronger variant that we call clique-circLWE where there are k keys each of which is encrypted with all other keys, under our basic circLWE assumption.
3. We prove the security of several representative FHE schemes [3,10,25,31,36] under circLWE.
4. We observe that the LWE circular security assumptions underlying the encryption schemes of [10,31] are essentially the same, namely circLWE (see Sect. 3 for a precise definition). This is a testament to the robust applicability of the assumption, and is enabled by an elegant perspective on GSW ciphertexts (originally developed in a sequence of talks by the first author as well as in [51]).
5. Even within a single encryption scheme (such as [10]), slightly different ways of encoding and encrypting the secret key seem to require different assump-

tions. Nevertheless, we show that circLWE implies the circular security of both variants, a further testament to the robustness of circLWE.
6. As an interesting direction for future research, we pose the question of whether one can show a worst-case to average-case reduction for circLWE, possibly under non-standard worst-case lattice assumptions.

The long-term challenge of this line of research is to prove circLWE from the standard LWE assumption or even the worst-case hardness of lattice problems. However, if this problem turns out to be intractable, we advocate making progress towards it by, e.g. showing a worst-case to average-case reduction starting from potentially new worst-case lattice assumptions.

To conclude, we clarify the *non-goals* of this paper. The quest to systematize the study of circular security of FHE schemes, as initiated in this paper, is a fundamental theoretical quest, one that will shed light on the security of essentially all the FHE constructions. Keeping this in mind, we study a subset of the FHE schemes, the foundational ones, focusing on LWE-based constructions, and, occasionally, making small changes to the schemes as required by our proofs.[3] We note that our results are not *directly* applicable to Ring-LWE and other variants of practical interest. Indeed, extending our results to a wider class of FHE schemes requires additional research. In particular, it requires not only adapting the results and proofs presented in this paper to the more challenging ring setting, but also investigate types of circular security information (e.g., automorphisms keys) that are specific to Ring-LWE. Still, these are extensions that can hopefully be informed by the techniques in our paper, and we leave them as an open problem.

Related Work: Circular-Secure Encryption Schemes. Constructing a circular secure encryption scheme was open until the work of Boneh, Halevi, Hamburg and Ostrovsky [9]. Several constructions have appeared since then under essentially all standard cryptographic assumptions [4,6,13,14]. None of these results, however, seem to imply the sort of circular security needed for FHE schemes.

Related Work: Counterexamples to Circular Security. A separate line of research has tried to extend the trivial counterexample from the first paragraph of the introduction to more demanding settings. For example, is there a *bit-wise* encryption scheme that is semantically secure yet circular *insecure*? Is there an encryption scheme that is insecure in the presence of key cycles of length ≥ 2? Neither question seems to have an obvious answer, yet we know sophisticated constructions that demonstrate that the answer to both is "yes" [7,34,35,42,43].

Organization of the Paper. The rest of the paper is organized as follows. In Sect. 2, we provide basic definitions and notation. Next, in Sect. 3, we formulate a number of conjectures that capture a circular secure variant of the LWE

[3] For example, we may use discrete Gaussian encryption noise with larger parameters than the original papers, or slightly different error distributions. These should be interpreted as artifacts of our proof techniques.

problem, and provide reductions between them. Finally, in Sect. 4, we analyze a representative set of FHE schemes.

2 Preliminaries

We write $\mathbb{Z}$ and $\mathbb{R}$ for the sets of integers and real numbers, respectively, and $\mathbb{Z}_p, \mathbb{R}_p$ for the integer and real numbers modulo $p \in \mathbb{Z}$, typically represented as values in the interval $[0, p)$ or $[-p/2, p/2)$. We write $\lfloor x \rfloor, \lceil x \rceil$ or $\lfloor x \rceil$ for the result of rounding $x \in \mathbb{R}$ down, up or to the closest integer, rounding up in case of a tie. We also define the *modulus switching* operation

$$\lfloor x \rceil_q = \left\lfloor \tfrac{q}{p} \cdot x \right\rceil$$

which maps any $\mathbf{x} \in \mathbb{Z}_p$ (or, more generally $\mathbf{x} \in \mathbb{R}_p$) to an element of $\mathbb{Z}_q$ (resp. $\mathbb{R}_q$). The input modulus p is implicitly defined by the domain of $x \in \mathbb{R}_p$, and we write $\lfloor x \pmod p \rceil_q$ when p is not clear from the context. We let $\mathbb{R}^+$, resp. $\mathbb{Z}^+$, denote the set of all non-negative reals, resp. integers.

We use bold lowercase letters $\mathbf{x}, \mathbf{y}$ for (column) vectors, and bold uppercase $\mathbf{X}, \mathbf{Y}$ for matrices. The transpose of a matrix $\mathbf{X}$ is denoted by $\mathbf{X}^t$. Row vectors are written using matrix transpose notation $\mathbf{x}^t$. We write $[\mathbf{X}_1, \ldots, \mathbf{X}_n]$ for (horizontal) concatenation of matrices with the same number of rows, and $(\mathbf{X}_1, \ldots, \mathbf{X}_n) = [\mathbf{X}_1^t, \ldots, \mathbf{X}_n^t]^t$ for (vertical) stacking of matrices. Unless stated otherwise, the coordinates of a vector are denoted by $\mathbf{x} = (x_1, \ldots, x_n)$. Similarly, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ for the columns of a matrix.

The inner product of two vectors is written either as $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_i x_i y_i$, or using matrix transpose notation $\mathbf{x}^t \cdot \mathbf{y}$. The tensor product between two matrices $\mathbf{X} \in \mathbb{Z}^{n_0 \times k_0}$ and $\mathbf{Y} \in \mathbb{Z}^{n_1 \times k_1}$ is the matrix $\mathbf{X} \otimes \mathbf{Y} \in \mathbb{Z}^{n_0 n_1 \times k_0 k_1}$ obtained by replacing each entry $x_{i,j}$ with a scaled copy $x_{i,j} \cdot \mathbf{Y}$ of $\mathbf{Y}$.

We say that $\mathbf{x}$ is a short vector in $\mathbb{Z}^n$ or $\mathbb{R}^n$ if $\|\mathbf{x}\|$ is small, with respect to some norm, e.g., $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$, or $\|\mathbf{x}\|_\infty = \max_i |x_i|$. All the statements in this paper can be analyzed using any choice of norm, producing essentially the same results, with a small difference in the concrete norm bound. By default, when we use the expression $\|\mathbf{x}\|$ without further qualifiers, we will mean the Euclidean norm of $\mathbf{x}$.

The Gaussian Distribution. The Gaussian, or the normal, distribution over $\mathbb{R}^n$, centered at $\mathbf{c} \in \mathbb{R}^n$ and parameterized by a standard deviation $\sigma \in \mathbb{R}^+$, is defined by the following probability density function:

$$\forall \mathbf{x} \in \mathbb{R}^n : \quad \mathcal{N}_\sigma(\mathbf{x}) = \frac{1}{\sigma} \cdot e^{-\pi ||\mathbf{x}-\mathbf{c}||^2/\sigma^2}$$

Similarly, the discrete Gaussian distribution χ_σ is defined as the restriction of $\mathcal{N}_\sigma$ to $\mathbb{Z}^n$, i.e., the discrete random variable over $\mathbb{Z}^n$ that outputs $\mathbf{x} \in \mathbb{Z}^n$ with probability $\mathcal{N}_\sigma(\mathbf{x}) / \sum_{\mathbf{y} \in \mathbb{Z}^n} \mathcal{N}_\sigma(\mathbf{y})$.

Cryptographic Primitives. All asymptotic statements are with respect to a security parameter λ, which is implicitly given as input to all algorithms and associated sets. So, for example, a key generation algorithm Gen is simply defined as an efficiently samplable distribution over a set of keys $\mathcal{K}$. By this we mean that $\mathsf{Gen}(\lambda)$ is a probabilistic algorithm that, on input λ, runs in time polynomial in λ, and outputs a key from a set of bit-strings $\mathcal{K}(\lambda)$ which may also depend on λ. Events (e.g., describing security or correctness properties) are similarly parametrized by λ, and the probability of an event defines a function $f(\lambda) \in [0,1]$. A probability (function) $f(\lambda)$ is *negligible* if $f(\lambda) < 1/\lambda^c$ for every constant c and all large enough values of λ. A probability $f(\lambda)$ is *overwhelming* if $1 - f(\lambda)$ is negligible.

Definition 1. *A private key encryption scheme* $\mathsf{SKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}$*, key space* $\mathcal{K}$ *and ciphertext space* $\mathcal{C}$*, is a triple of (probabilistic polynomial time) algorithms* $\mathsf{Gen}\colon \mathcal{K}$ *(key generation),* $\mathsf{Enc}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ *(encryption) and* $\mathsf{Dec}\colon \mathcal{K} \times \mathcal{C} \to \mathcal{M}$ *(decryption), such that* $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$ *for all messages* $m \in \mathcal{M}$ *and keys* $k \in \mathcal{K}$*. The correctness condition* $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$ *can be relaxed to hold only with overwhelming probability, over the choice of the key* k *and the encryption randomness.*

Homomorphic public-key encryption schemes can be constructed generically from any homomorphic private-key encryption scheme [57]. So, for simplicity, *we focus on the case of secret-key encryption.* All definitions and constructions are easily extended to the public-key setting. Still, we define security of private-key encryption in the presence of some public information $\mathsf{Pub}(k)$ about the secret key. This is useful to model the evaluation key used by some homomorphic operations, as well as bootstrapping. It can also be used to model certain forms of leakage resilience, like circular security. The standard notion of security for private-key encryption is obtained by letting Pub output nothing.

Definition 2. *Let* $\mathsf{SKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a private-key encryption scheme, and* $\mathsf{Pub}\colon \mathcal{K} \to \mathcal{P}$ *a (possibly randomized, efficiently computable) function from the set of keys* $\mathcal{K}$ *to some set* $\mathcal{P}$*. The scheme* SKE *satisfies* indistinguishability under chosen plaintext attack *(IND-CPA security for short) in the presence of public information* Pub *if any efficient (probabilistic polynomial time) adversary* $\mathcal{A}$ *can only achieve a negligible advantage in the following game, parametrized by a bit* $b \in \{0,1\}$*: after generating parameters* $k \leftarrow \mathsf{Gen}$*,* $p \leftarrow \mathsf{Pub}(k)$*, the adversary* $b' \leftarrow \mathcal{A}^{O_b(\cdot,\cdot)}(p)$ *is run on input* p *and with access to a (probabilistic) oracle* $O_b(m_0, m_1) = \mathsf{Enc}_k(m_b)$ *that on input a pair of messages* $m_0, m_1 \in \mathcal{M}$*, computes the encryption of the message selected by the bit* b*. Upon termination, the adversary outputs a bit* b'*, with the goal of correctly guessing the value of* b*. The adversary's advantage*[4] *is defined as* $\mathsf{Adv}(\mathcal{A}) = |\Pr\{b' = 1 \mid b = 0\} - \Pr\{b' = 1 \mid b = 1\}|$.

[4] More refined notions of advantage that better capture the quantitative notion of "bit-security" are proposed in [53]. Here we use the traditional definition of advantage, which is simpler, and still adequate for our purposes.

As an important special case, we consider *circular-secure* (private-key) encryption schemes, i.e., schemes satisfying security with respect to adversaries that are given, as auxiliary information, an encryption of the secret key k under itself. Notice that in order to encrypt a key k, one must first encode k as a sequence $\psi(k) \in \mathcal{M}^w$ of elements from the message space. The encryption function is extended to $\mathcal{M}^w$ componentwise, setting $\mathsf{Enc}_k(m_1, \ldots, m_w) = (\mathsf{Enc}_k(m_1), \ldots, \mathsf{Enc}_k(m_w))$.

Definition 3. *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Pub})$ *be an encryption scheme with message space* $\mathcal{M}$*, key space* $\mathcal{K}$ *and public information* Pub*, and let* $\psi\colon \mathcal{K} \to \mathcal{M}^w$ *be an (efficiently computable) encoding function. The encryption scheme is* ψ*-circular IND-CPA secure in the presence of* Pub *if it is IND-CPA secure with respect to the extended public information*[5] $\widehat{\mathsf{Pub}}(k) = (\mathsf{Pub}(k), \mathsf{Enc}_k(\psi(k)))$.

2.1 The Learning with Errors Problem (with Side Information)

The Learning With Errors (LWE) problem is an injective version of the Short Integer Solution (SIS) problem [2,52]. Its (average-case) computational hardness was proved by Regev in [56] based on the conjectured hardness of solving several standard lattice problems in the worst case, with further improvements in [15,54].

In this paper, we use the following matrix version [30,55] of LWE. The definition is parametrized by a secret distribution $\mathcal{S}$, which is typically set to either the uniform distribution over $\mathcal{S} = \mathbb{Z}_q^n$, or the same as the error distribution $\mathcal{S} = \chi^n$, which is equivalent to uniform secrets by the results of [4,46], or to uniformly random binary vectors $\mathcal{S} = \{0,1\}^n$. The latter choice is often made for efficiency reasons, and is theoretically justified by the results of [11,15,32,48].

Definition 4 (Learning With Errors (LWE) Distribution). *The LWE distribution with parameters* n, k, w, q*, a secret distribution* $\mathcal{S}$ *over* $\mathbb{Z}$ *and an error distribution* χ *over* $\mathbb{Z}^w$*, is given by* $[\mathbf{A}, \mathbf{AS} + \mathbf{E}]$ *where* $\mathbf{A} \leftarrow \mathbb{Z}_q^{w \times n}$*,* $\mathbf{S} \leftarrow \mathcal{S}^{n \times k}$ *and* $\mathbf{E} \leftarrow \chi^{w \times k}$.

In order to study the security of homomorphic encryption schemes, we parameterize the LWE hardness assumption by a public information function Pub, similarly to Definition 2. The standard LWE assumption is given by setting $\mathsf{Pub}(\mathbf{S}) = \perp$.

Definition 5 *Let* $\mathsf{Pub}(\mathbf{S})$ *be any efficiently computable, possibly randomized function of the LWE secret* $\mathbf{S}$*. The* Decisional LWE problem *with public information* Pub *is the computational problem of distinguishing between the following two distributions:*

[5] Here we are starting from an encryption scheme that already includes a Pub function (e.g., to provide a public key or other side information), and extend it to include an encryption cycle. When starting from a simple private-key encryption scheme, $\mathsf{Pub}(k)$ outputs nothing, and $\widehat{\mathsf{Pub}}(k) = \mathsf{Enc}_k(\psi(k))$ is just a circular encryption of the key.

- $(\mathsf{Pub}(\mathbf{S}), [\mathbf{A}, \mathbf{AS} + \mathbf{E}])$ *where* $\mathbf{A} \leftarrow \mathbb{Z}_q^{w \times n}$, $\mathbf{S} \leftarrow \mathcal{S}^{n \times k}$ *and* $\mathbf{E} \leftarrow \chi^{w \times k}$.
- $(\mathsf{Pub}(\mathbf{S}), [\mathbf{A}, \mathbf{B}])$ *where* $\mathbf{A} \leftarrow \mathbb{Z}_q^{w \times n}$, $\mathbf{S} \leftarrow \mathcal{S}^{n \times k}$ *and* $\mathbf{B} \leftarrow \mathbb{Z}_q^{w \times k}$.

The decisional LWE assumption postulates that the decisional LWE problem is hard to solve with non-negligible advantage for any probabilistic polynomial-time distinguisher.

One may also give a slightly stronger definition which requires the distribution $(\mathsf{Pub}(\mathbf{S}), [\mathbf{A}, \mathbf{AS} + \mathbf{E}])$ to be indistinguishable from $(\mathbf{P}, [\mathbf{A}, \mathbf{B}])$, where $\mathbf{P}, \mathbf{A}, \mathbf{B}$ are all chosen uniformly at random. Most of the results in this paper hold also for this stronger definition, but this is not needed for the application to circular security and fully homomorphic encryption.

Note that when $\mathsf{Pub}(\mathbf{S}) = \perp$ (or, more generally, when Pub provides independent public information $[\mathsf{Pub}(\mathbf{s}_1), \ldots, \mathsf{Pub}(\mathbf{s}_k)]$ about each column of $\mathbf{S}$), one may assume $k = 1$, and prove the hardness for any k by a standard hybrid argument [55]. However, the same does not generally hold true in the presence of global information $\mathsf{Pub}(\mathbf{S})$ about the whole secret matrix $\mathbf{S}$.

On the LWE Secret and Noise Distributions. Since LWE is an average-case problem, its computational hardness depends on the specific distributions $\mathcal{S}$ and χ used in the definition. Still, the hardness of LWE is fairly robust both with respect to the secret key distribution $\mathcal{S}$ (via *leakage resilience* results [11, 15, 32, 48]) and the error distribution χ. The main requirement on the noise distribution χ is that it should output small numbers, and the strength of the hardness assumption is often quantified by the ratio $q/|\chi|$ between the errors and the ciphertext modulus.

We state below a useful theorem on LWE with binary secrets [48].

Lemma 1 (Hardness of Binary-Secret LWE, adapted from [48]). *Assume the hardness of the decisional LWE problem with modulus*[6] q*, uniform secrets* $\mathbf{s} \in \mathbb{Z}_q^{n'}$*,* $n+1$ *samples, and discrete Gaussian noise* χ_σ *is hard, for some* $q \leq 2^{n^{O(1)}}$*,* $\sigma \geq \omega(\sqrt{\log n})$*, and* $n \geq 2n' \log_2 q$*. Then, the Decisional LWE problem with binary secrets* $\mathbf{s} \in \{0,1\}^n$*, polynomially many samples, and discrete Gaussian noise* $\chi_{\sigma \cdot \sqrt{n'}}$ *is also hard.*

A common technique to "smooth out" differences between error distributions is *noise flooding*, i.e., the addition of random noise r to the LWE matrix $\mathbf{B}$, so that the error becomes $\chi + r$. If r is random and sufficiently larger than χ, then $\chi' = \chi + r$ becomes essentially independent of χ. Clearly, this technique (as well as many other methods to reduce between different versions of LWE) has the side effect of increasing the amount of error. However, one can usually compensate for the larger error χ' by using a correspondingly larger modulus q.

[6] To be precise, [48] proves this result for odd moduli q, and then explains in [48, Footnote 2] how to adapt the result to even moduli using modulo switching. Technically using modulus switching requires a small increase in the LWE modulus q when going from uniform to binary secrets, but, for simplicity, we ignore this technicality.

We state a version of the noise-flooding lemma below, adapted from [5] (where it is stated with respect to the uniform, rather than the Gaussian, distribution).

Lemma 2 (Noise-flooding Lemma, adapted from [5, Lemma 1]). *Let $n \in \mathbb{Z}$ and let $\mathcal{E}$ be a set of vectors $\mathbb{R}^n$. Let $\mathbf{e} \in \mathcal{E}$. Then, the statistical distance between the distributions $\mathcal{N}_\sigma$ and $\mathbf{e} + \mathcal{N}_\sigma$ is $O(||\mathbf{e}||/\sigma)$. Similarly for the discrete Gaussian distribution χ_σ, as long as $\sigma = \omega(\sqrt{\log n})$.*

2.2 LWE Encryption

The LWE problem can be used to define a family of randomized functions, parametrized by n, k, w, q and indexed by the keys $\mathbf{S} \leftarrow \mathcal{S}^{n \times k}$, which, on input a matrix $\mathbf{X} \in \mathbb{Z}_q^{w \times k}$, chooses $\mathbf{A} \leftarrow \mathbb{Z}_q^{w \times n}$ and $\mathbf{E} \leftarrow \chi^{w \times k}$ at random, and outputs

$$\mathsf{LWE}_{\mathbf{S}}(\mathbf{X}; \mathbf{A}, \mathbf{E}) \stackrel{\text{def}}{=} [\mathbf{A}, \mathbf{X} + \mathbf{E} - \mathbf{AS}] \in \mathbb{Z}_q^{w \times (n+k)}. \tag{1}$$

We write $\mathsf{LWE}_{\mathbf{S}}(\mathbf{X})$ for the output distribution obtained by choosing $\mathbf{A}$ and $\mathbf{E}$ at random, and computing $\mathsf{LWE}_{\mathbf{S}}(\mathbf{X}; \mathbf{A}, \mathbf{E})$. Similarly, one can define an (approximate) inversion algorithm that, given a key $\mathbf{S}$ and a ciphertext $[\mathbf{A}, \mathbf{B}] \in \mathbb{Z}_q^{w \times (n+k)}$, outputs:

$$\mathsf{LWE}_{\mathbf{S}}^{-1}([\mathbf{A}, \mathbf{B}]) \stackrel{\text{def}}{=} [\mathbf{A}, \mathbf{B}] \cdot \begin{bmatrix} \mathbf{S} \\ \mathbf{I} \end{bmatrix} = \mathbf{AS} + \mathbf{B} \in \mathbb{Z}_q^{w \times k}. \tag{2}$$

Notice that $(\mathsf{LWE}, \mathsf{LWE}^{-1})$ is not quite an encryption scheme because it only satisfies an approximate version of the correctness condition

$$\mathsf{LWE}_{\mathbf{S}}^{-1}(\mathsf{LWE}_{\mathbf{S}}(\mathbf{X})) = \mathbf{AS} + \mathbf{B} = \mathbf{X} + \mathbf{E} \approx \mathbf{X}$$

up to a small additive error $\mathbf{E}$. In order to get a proper encryption scheme, LWE is combined with an error correcting code, as described next.

For simplicity, we focus on linear codes defined by a single (so-called *gadget*) vector $\mathbf{g} \in \mathbb{Z}_p^w$, where p is a plaintext modulus possibly different from q, as these are the codes most commonly used in lattice-based cryptography. Using $\mathbf{g}$ as a gadget, a message $\mathbf{M} \in \mathbb{Z}_p^{h \times k}$ is encoded componentwise as $\mathbf{M} \otimes \mathbf{g}$, i.e., by replacing each entry $m_{i,j}$ by the vector $m_{i,j} \cdot \mathbf{g}$. Sometimes it is convenient to express the encoding $\mathbf{M} \otimes \mathbf{g}$ as a matrix product, rather than a tensor. This is easily done by letting $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}$, and observing that

$$\mathbf{G} \cdot \mathbf{M} = (\mathbf{I} \otimes \mathbf{g}) \cdot (\mathbf{M} \otimes (1)) = (\mathbf{IM}) \otimes (\mathbf{g} \cdot (1)) = \mathbf{M} \otimes \mathbf{g}.$$

(Notice that when using matrix product, the message $\mathbf{M}$ is multiplied by $\mathbf{G}$ on the left.)

Gadget vectors $\mathbf{g}$ are required to satisfy the primitivity condition $\gcd(p, \mathbf{g}) = 1$, so that the encoding function is injective, and $m_{i,j}$ can be recovered from $m_{i,j} \cdot \mathbf{g}$. The encoding is rounded to a matrix $\lfloor \mathbf{M} \otimes \mathbf{g} \rceil_q$ in $\mathbb{Z}_q^{hw \times k}$ before applying the LWE function, producing the ciphertext

$$\mathbf{g}\text{-}\mathsf{LWE}_{q,\mathbf{S}}(\mathbf{M}; \mathbf{A}, \mathbf{E}) \stackrel{\text{def}}{=} \mathsf{LWE}_{\mathbf{S}}\left(\lfloor \mathbf{M} \otimes \mathbf{g} \rceil_q\right) \in \mathbb{Z}_q^{hw \times (n+k)}. \tag{3}$$

For brevity, we omit the ciphertext modulus when $q = p$ and the gadget vector when $\mathbf{g} = (1)$, and simply write $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}}$ or $\mathsf{LWE}_{q,\mathbf{S}}$, noting that this is consistent with the notation $\mathsf{LWE}_{\mathbf{S}}$ used for the plain (unencoded) LWE function defined in Eq. (1). We also write $\mathbf{g}\text{-}\mathsf{LWE}_{q,\mathbf{S}}(\mathbf{M};\mathcal{E})$ for the set of $\mathbf{g}$-LWE ciphertexts with error in the set $\mathcal{E}$ (and any matrix $\mathbf{A}$), or an arbitrary element of that set.

We will not be concerned with the decryption algorithm, as it plays no role in the definition of security.[7] For completeness, we only briefly mention that ciphertexts $\mathbf{C} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}}(\mathbf{M})$ can be decrypted by computing the matrix $\mathbf{M} \in \mathbb{Z}_p^{h\times k}$ such that the encoding $\mathbf{M}\otimes\mathbf{g}$ is closest to $\mathbf{X} = \frac{p}{q}\cdot \mathsf{LWE}_{\mathbf{S}}^{-1}(\mathbf{C})$.

Besides *encoding*, and *decoding* (as used for encryption and decryption), gadget vectors $\mathbf{g} \in \mathbb{Z}_p^w$ have one more *inversion* algorithm that on input any $x \in \mathbb{Z}_p$ outputs a *short* integer (row) vector $\mathbf{g}^{-t}(x) \in \mathbb{Z}^{1\times w}$ such that $\mathbf{g}^{-t}(x)\cdot\mathbf{g} = x \pmod p$. More generally, for any $\mathbf{M} \in \mathbb{Z}_p^{h\times k}$ and $\mathbf{T} \in \mathbb{Z}_p^{l\times h}$, the encoding and inversion operations satisfy

$$(\mathbf{g}^{-t}(\mathbf{T}))\cdot(\mathbf{M}\otimes\mathbf{g}) = \mathbf{TM} \pmod p$$

where, as usual, $\mathbf{g}^{-t}(\cdot)$ is extended to vectors and matrices componentwise. Sometimes it is convenient to use the output of $\mathbf{g}^{-t}(x)$ in column form, which we write as $\mathbf{g}^{-1}(x)$. More generally, for any matrix $\mathbf{X}$, we have $\mathbf{g}^{-1}(\mathbf{X}) \stackrel{\text{def}}{=} (\mathbf{g}^{-t}(\mathbf{X}^t))^t$. Following [3,51], this function is used to define the following homomorphic operation that plays a fundamental role in our presentation of LWE encryption and FHE.

Definition 6 *For any gadget* $\mathbf{g} \pmod p$*, message* $\mathbf{M} \pmod p$*, ciphertext* $\mathbf{C} \in \mathbf{g}\text{-}\mathsf{LWE}_{q,\mathbf{S}}(\mathbf{M})$ *and matrix* $\mathbf{T} \in \mathbb{Z}_p^{l\times h}$*, define the* gadget product

$$\mathbf{T}\odot\mathbf{C} \stackrel{def}{=} \mathbf{g}^{-t}(\mathbf{T})\cdot\mathbf{C} \in \mathsf{LWE}_{q,\mathbf{S}}(\mathbf{TM}) \tag{4}$$

where the vector $\mathbf{g}$ *is implicitly specified by the type of ciphertext* $\mathbf{C}$*.*

This operation multiplies a message $\mathbf{M}$ encrypted under $\mathbf{g}$-LWE by a matrix $\mathbf{T}$, producing as a result an *unencoded* LWE encryption of $\mathbf{TM}$. Products encoded under any (possibly different) gadget $\mathbf{h} \pmod p$ can be computed as

$$\begin{aligned}\mathbf{T} \overset{\mathbf{h}}{\odot} \mathbf{C} &\stackrel{\text{def}}{=} (\mathbf{T}\otimes\mathbf{h})\odot\mathbf{C} \in \mathsf{LWE}_{q,\mathbf{S}}((\mathbf{T}\otimes\mathbf{h})\mathbf{M}) \\ &= \mathsf{LWE}_{q,\mathbf{S}}(\mathbf{TM}\otimes\mathbf{h}) = \mathbf{h}\text{-}\mathsf{LWE}_{q,\mathbf{S}}(\mathbf{TM}).\end{aligned}$$

As a special case, we write $\mathbf{T}\circledcirc\mathbf{C} \stackrel{\text{def}}{=} \mathbf{T}\overset{\mathbf{g}}{\circledcirc}\mathbf{C}$ for the operation of applying $\mathbf{T}$ to a $\mathbf{g}$-LWE ciphertext without changing the encoding gadget $\mathbf{g}$, and $\mathbf{h}\circ\mathbf{C} \stackrel{\text{def}}{=} \mathbf{I}\overset{\mathbf{h}}{\circledcirc}\mathbf{C}$ for the operation of changing the encoding gadget from $\mathbf{g}$ to $\mathbf{h}$ without modifying the message.

[7] This is for valid encryption schemes, satisfying the standard correctness condition $\mathsf{Dec}_s(\mathsf{Enc}_s(m)) = m$. For "approximate" encryption schemes where $\mathsf{Dec}_s(\mathsf{Enc}_s(m)) \approx m$, see [44,45]. We only consider valid encryption schemes in this paper.

2.3 Key Switching

As we show in this section, the gadget product operation $\odot$ can be used to define a "functional" key switching procedure that changes the key under which a ciphertext is encrypted, while at the same time multiplying the message by a given integer matrix $\mathbf{T}$ on the *right*. (Notice that this is different from the gadget product $\mathbf{T}\odot$, which multiplies the message by $\mathbf{T}$ on the *left*.) Plain key switching is a special case where $\mathbf{T} = \mathbf{I}$ is the identity function.

Definition 7. *For any two keys $\mathbf{S} \in \mathbb{Z}^{n\times k}$ and $\tilde{\mathbf{S}} \in \mathbb{Z}^{\tilde{n}\times\tilde{k}}$, gadget $\mathbf{g} \in \mathbb{Z}_q^w$, and matrix $\mathbf{T} \in \mathbb{Z}^{\tilde{k}\times k}$ with (small) integer entries, define the* switching key *generation algorithm*

$$\mathbf{g}\text{-}\mathsf{LWE}_{\tilde{\mathbf{S}}\to\mathbf{S}}(\mathbf{T};\mathbf{A},\mathbf{E}) \stackrel{def}{=} \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}}\left(\begin{bmatrix}\tilde{\mathbf{S}}\\ \mathbf{I}\end{bmatrix}\mathbf{T};\mathbf{A},\mathbf{E}\right), \tag{5}$$

and similarly for $\mathbf{g}\text{-}\mathsf{LWE}_{\tilde{\mathbf{S}}\to\mathbf{S}}(\mathbf{T})$, etc.

We refer to ciphertexts of the form $\mathbf{g}\text{-}\mathsf{LWE}_{\tilde{\mathbf{S}}\to\mathbf{S}}(\mathbf{T})$ as *switching keys*, as they can be used to map encryptions of $\mathbf{M}$ under $\tilde{\mathbf{S}}$ to encryptions of $\mathbf{MT}$ under $\mathbf{S}$ as shown in the next theorem. Indeed, they correspond exactly to switching keys in [10,12,16].

Theorem 1. *For any keys $\mathbf{S} \in \mathbb{Z}_{n\times k}$, $\tilde{\mathbf{S}} \in \mathbb{Z}_{\tilde{n}\times\tilde{k}}$, gadget $\mathbf{g} \in \mathbb{Z}_q^w$, matrix $\mathbf{T} \in \mathbb{Z}^{\tilde{k}\times k}$ with (small) integer entries, switching key*

$$\mathbf{W} \in \mathbf{g}\text{-}\mathsf{LWE}_{\tilde{\mathbf{S}}\to\mathbf{S}}(\mathbf{T};\mathcal{F}),$$

possibly different plaintext modulus p, gadget $\mathbf{h} \in \mathbb{Z}_p^{\tilde{w}}$, message $\mathbf{M} \in \mathbb{Z}_p^{\tilde{h}\times k}$, and ciphertext $\mathbf{C} \in \mathbf{h}\text{-}\mathsf{LWE}_{q,\tilde{\mathbf{S}}}(\mathbf{M};\mathcal{E})$ we have

$$\mathbf{C}\odot\mathbf{W} \in \mathbf{h}\text{-}\mathsf{LWE}_{q,\mathbf{S}}(\mathbf{MT};\mathcal{E}'),$$

where $\mathcal{E}' = \mathcal{E}\cdot\mathbf{T} + \mathbf{g}^{-t}(\mathbf{C})\,\mathcal{F}$.

Proof. This follows by a simple calculation:

$$\begin{aligned}
\mathbf{C}\odot\mathbf{W} &= \mathbf{C}\odot\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}}\left(\begin{bmatrix}\tilde{\mathbf{S}}\\ \mathbf{I}\end{bmatrix}\mathbf{T};\mathcal{F}\right)\\
&= \mathsf{LWE}_{q,\mathbf{S}}\left(\mathbf{C}\begin{bmatrix}\tilde{\mathbf{S}}\\ \mathbf{I}\end{bmatrix}\mathbf{T};\mathbf{g}^{-t}(\mathbf{C})\cdot\mathcal{F}\right)\\
&= \mathsf{LWE}_{q,\mathbf{S}}\left((\tfrac{q}{p}\mathbf{M}\otimes\mathbf{h}+\mathcal{E})\mathbf{T};\mathbf{g}^{-t}(\mathbf{C})\cdot\mathcal{F}\right)\\
&= \mathsf{LWE}_{q,\mathbf{S}}\left(\tfrac{q}{p}(\mathbf{MT})\otimes\mathbf{h};\mathcal{E}\cdot\mathbf{T}+\mathbf{g}^{-t}(\mathbf{C})\cdot\mathcal{F}\right)\\
&= \mathbf{h}\text{-}\mathsf{LWE}_{q,\mathbf{S}}(\mathbf{MT};\mathcal{E}')
\end{aligned}$$

where $\mathcal{E}' = \mathcal{E}\cdot\mathbf{T} + \mathbf{g}^{-t}(\mathbf{C})\cdot\mathcal{F}$ is small because $\mathcal{E}$, $\mathbf{T}$, $\mathbf{g}^{-t}(\mathbf{C})$ and $\mathcal{F}$ are all small.

Using the fact that switching keys are just regular **g**-LWE ciphertexts (of carefully crafted, key-dependent messages), it is easy to see that switching keys can be combined both by pointwise addition and function composition:

$$\mathbf{g}\text{-LWE}_{\mathbf{S}\to\mathbf{S}'}(\mathbf{T}_0) + \mathbf{g}\text{-LWE}_{\mathbf{S}\to\mathbf{S}'}(\mathbf{T}_1) = \mathbf{g}\text{-LWE}_{\mathbf{S}\to\mathbf{S}'}(\mathbf{T}_0 + \mathbf{T}_1)$$
$$\mathbf{g}\text{-LWE}_{\mathbf{S}'\to\mathbf{S}''}(\mathbf{T}_0) \circledcirc \mathbf{g}\text{-LWE}_{\mathbf{S}\to\mathbf{S}'}(\mathbf{T}_1) = \mathbf{g}\text{-LWE}_{\mathbf{S}\to\mathbf{S}''}(\mathbf{T}_0 \cdot \mathbf{T}_1)$$

This turns out to be a *very interesting* insight: GSW ciphertexts [31] *are* switching keys where the transformation $\mathbf{T} = m\mathbf{I}$ is a scalar matrix representing the message m, and the observation above shows how to do additive and multiplicative homomorphisms on GSW ciphertexts. For more details, we refer the reader to Sect. 4.2.

2.4 Gadgets

We conclude this section with a brief discussion of the gadget vectors most commonly used in lattice cryptography. We will be primarily concerned with the "power base" gadget

$$\mathbf{pow}(b) \stackrel{\text{def}}{=} (1, b, b^2, \ldots, b^{w-1}) \in \mathbb{Z}_p^w$$

for $p = b^w$, possibly equal to the ciphertext modulus q. In fact, both for simplicity and historical reasons, most theoretical papers use the "power-of-two" gadget $\mathbf{pow}(2)$, i.e., the special case where $b = 2$. Efficient (bounded distance) decoding algorithms (used for decryption) are given in [50] (for $p = b^w$) and [27] (for arbitrary p). More relevant for this work is the gadget inversion algorithm $\mathbf{pow}(b)^{-t}(x)$ which outputs the (signed) base b representation of x. Randomized (subgaussian) inversion algorithms, with somewhat better average error growth, are given in [3,28].

The power gadget $\mathbf{pow}(b)$ is most commonly used with a plaintext modulus $p = q$ equal to the LWE ciphertext modulus. Another important special case is when $b = p$, and $\mathbf{g} = (1) \in \mathbb{Z}_p^1$ is the trivial vector. This is typically used with a fixed plaintext modulus p much smaller than the ciphertext modulus q. (E.g., $p = 2$ to encrypt single bits $m \in \{0, 1\}$.) The decoding algorithm for this gadget is just a simple rounding operation to the closest integer modulo p. Inversion is just as simple: $\mathbf{g}^{-t}(x) = x \bmod p$ outputs the signed integer representative of x in $[-p/2, \ldots, p/2)$, or a centered binary random variable taking as possible value the representative(s) of $x \pmod p$ in $(-p, \ldots, p)$.

Several other gadgets are often used in practice, to provide better efficiency, parallelism, or useful tradeoffs between ciphertext size and error growth. These include power gadgets $\mathbf{pow}(b)$ with a large base $b \approx \sqrt{p}$ or $b \approx p^{1/3}$ (so that $\mathbf{g}$ has only two or three coordinates), and the Residue Number System (RNS) gadget, which uses a highly composite $p = \prod_i p_i$ and represents integers $x \in \mathbb{Z}_p$ as $(x \bmod p_1, \ldots, x \bmod p_k)$ using the Chinese Reminder Theorem. These are also most commonly used with $p = q$.

3 Circular LWE Conjectures

We describe the *quadratic circular LWE* assumption, called circLWE, using notation from Sect. 2. Let $\mathbf{g}$ be any gadget in dimension w and define the quadratic function

$$\psi_{\mathbf{g}}(\mathbf{s}) = \mathbf{g}^{-1}(\mathbf{s}, 1) \otimes \mathbf{g}^{-1}(\mathbf{s}, 1) \in \mathbb{Z}_q^{((n+1)w)^2}$$

The vector $\mathbf{g}$ is usually the power-of-two gadget $\mathbf{pow}(2)$ and $w = \lceil \log_2 q \rceil$ We omit the subscript and simply write $\psi(\mathbf{s})$ instead of $\psi_{\mathbf{g}}(\mathbf{s})$ when $\mathbf{g}$ is clear from the context, or unimportant.

The $\mathbf{g}$-circLWE assumption says that the decisional LWE problem with public information $\mathsf{Pub}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi_{\mathbf{g}}(\mathbf{s}))$ (see Definition 5) is computationally hard. More specifically, parametrizing the definition by the LWE noise distributions used to compute $\mathsf{Pub}(\mathbf{s})$ and the LWE samples, $\mathbf{g}\text{-}\mathsf{circLWE}[\xi, \Xi]$ says that no probabilistic polynomial-time distinguisher $\mathcal{D}$ can distinguish between the following two distributions with non-negligible advantage, for any $\ell = \mathsf{poly}(n)$:

- $(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}); \Xi), \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0}^{\ell}; \xi))$ where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$.
- $(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}); \Xi), \mathbf{U})$ where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{U} \leftarrow \mathbb{Z}_q^{\ell w \times (n+1)}$.

We omit ξ and Ξ when they are both equal to the standard LWE error[8] (discrete Gaussian) distribution $\chi_{\sqrt{n}}$. Naturally, $\mathbf{g}$-circLWE reduces to $\mathbf{g}\text{-}\mathsf{circLWE}[\chi_\sigma, \chi_{\sigma'}]$ for larger $\sigma > \sqrt{n}$ simply by adding more Gaussian noise to both $\mathsf{Pub}(\mathbf{s})$ and the LWE samples.

As above, we will refer to the $\mathbf{g}$-circLWE assumption when $\mathbf{g} = \mathbf{pow}(2)$ as simply circLWE. Our main conjecture is that the circLWE assumption is true under the decisional LWE assumption.

> **Main Conjecture 1** *The* circLWE *assumption (i.e., Definition 5 with side information* $\mathsf{Pub}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi_{\mathbf{g}}(\mathbf{s}))$*) is true assuming that the decisional LWE assumption (i.e., Definition 5 with* $\mathsf{Pub}(\mathbf{s}) = \perp$*) is true.*

3.1 How About *Linear Circular LWE?*

One may wonder about a variant of the above assumption that asks to encrypt only the bits of the secret key. That is, letting

$$\phi(\mathbf{s}) := \phi_{\mathbf{g}}(\mathbf{s}) = \mathbf{g}^{-1}(\mathbf{s}) \in \mathbb{Z}_q^{(n+1)w} \ ,$$

is it the case that LWE with side information $\mathsf{Pub}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\phi(\mathbf{s}))$ is hard, i.e.,

$$(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\phi(\mathbf{s})), \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0}^{\ell})) \approx_c (\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\phi(\mathbf{s})), \mathbf{U})?$$

[8] This is the smallest error for which the LWE problem is known to be as hard as worst case lattice problems [56].

We call this the linear circular LWE assumption. We show that the linear circular LWE assumption is true under the decisional LWE assumption. We remark that variants of this statement were known when the key is not decomposed into bits (i.e., when $\phi(\mathbf{s}) = \mathbf{s}$ and $\mathsf{Pub}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{s})$). E.g., [4] shows that assuming LWE, Regev's encryption is secure given as auxiliary encryption any affine function over the secret key. However, our result does not follow from [4] because the binary decomposition function $\phi_{\mathbf{g}}(\mathbf{s}) = \mathbf{g}^{-1}(\mathbf{s})$ is not linear (or even affine) in $\mathbf{s}$. So, to the best of our knowledge, the statement below is new.

Theorem 2. *Let $q = 2^k$ be a power of 2, and $\mathbf{g} = (1, 2, 4, \ldots, 2^{k-1})$ be the powers-of-two gadget vector. The linear circular LWE assumption (i.e., Definition 5 with public information $\mathsf{Pub}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\phi_{\mathbf{g}}(\mathbf{s}))$) with secret dimension $n \geq 2n' \log_2 q$ and discrete Gaussian noise distribution $\xi = \Xi = \chi_{\sigma\sqrt{n}}$ is true assuming the hardness of decisional LWE (i.e., Definition 5 with $\mathsf{Pub}(\mathbf{s}) = \perp$) with secret dimension n' and discrete Gaussian noise χ_σ.*

Proof. We actually prove a slightly stronger property, showing that

$$(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\phi(\mathbf{s})), \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0})) \in \mathbb{Z}_q^{(kn+kw)\times(n+1)}$$

is indistinguishable from a uniformly random matrix $(\mathbf{U}_0, \mathbf{U}_1)$ modulo q, where the $\mathbf{g}$-LWE ciphertexts use discrete Gaussian noise $\chi_{\sigma\sqrt{n}}$. We proceed in steps, giving a sequence of reductions, starting from the standard decisional LWE problem and ending with the linear circular LWE assumption. First, we use Lemma 1 to reduce the standard decisional LWE problem (with a uniformly random secret $\mathbf{s} \in \mathbb{Z}_q^{n'}$ and discrete Gaussian noise χ_σ) to the decisional LWE problem with a random binary secret $\mathbf{s}_0 \in \{0,1\}^n$, modulus q, and $kn + kw$ many samples. This is the only step of our proof that changes/increases the LWE dimension and noise. All the remaining steps preserve the error distribution and are very efficient. So, from this point on, we fix the LWE noise vector $\mathbf{e} \leftarrow \chi_{\sigma\sqrt{n}}^{kn+kw}$ and omit it from the notation.

Next, we reduce LWE with binary secret to the problem of distinguishing

$$\begin{bmatrix} \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_0}(\mathbf{s}_0) \\ \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_0}(\mathbf{0}) \end{bmatrix} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_0}\left(\begin{bmatrix} \mathbf{s}_0 \\ \mathbf{0} \end{bmatrix}\right) = [\mathbf{A}', \mathbf{b}] \tag{6}$$

from the uniform distribution over $\mathbb{Z}_q$. Let $[\mathbf{A}, \mathbf{b}]$ be the input LWE distribution (with binary secret $\mathbf{s}_0$ and error $\mathbf{e}$), and map it to $[\mathbf{A}', \mathbf{b}] = [\mathbf{A}, \mathbf{b}] + \left[\begin{bmatrix} \mathbf{I} \\ \mathbf{O} \end{bmatrix} \otimes \mathbf{g}, \mathbf{0}\right]$. This transformation clearly maps the uniform distribution to the uniform distribution. On the other hand, $\left[\begin{bmatrix} \mathbf{I} \\ \mathbf{O} \end{bmatrix} \otimes \mathbf{g}, \mathbf{0}\right]$ is a noiseless $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_0}$ encryption of $(\mathbf{s}_0, \mathbf{0})$. So, if $[\mathbf{A}, \mathbf{b}]$ is an LWE instance, then $[\mathbf{A}', \mathbf{b}]$ is a random $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_0}$ encryption of $(\mathbf{s}_0, \mathbf{0})$, i.e., it is distributed according to (6), with the same error $\mathbf{e}$.

Next map $\mathbf{s}_0$ to a random key $\mathbf{s} = \sum_i 2^i \mathbf{s}_i \in \mathbb{Z}_q^n$ with binary decomposition $\mathbf{s}_i \in \{0,1\}^n$. We want to map (6) to

$$\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_0}\left(\begin{bmatrix}\mathbf{g}^{-1}(\mathbf{s})\\ \mathbf{0}\end{bmatrix}\right) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_0}\left(\begin{bmatrix}\mathbf{s}_0\\ \mathbf{s}_1\\ \vdots\\ \mathbf{s}_k\\ \mathbf{0}\end{bmatrix}\right) = [\mathbf{A}', \mathbf{b}']. \tag{7}$$

This is done simply by picking $\mathbf{s}_1, \ldots, \mathbf{s}_k \in \{0,1\}^n$ uniformly at random, and adding $\mathbf{s}_i \otimes \mathbf{g}$ to the last column of (6), setting $\mathbf{b}' = \mathbf{b} + (\mathbf{0}, \mathbf{s}_1, \ldots, \mathbf{s}_k, \mathbf{0}) \otimes \mathbf{g}$. This is correct because $[\mathbf{O}, \mathbf{s}_i \otimes \mathbf{g}]$ is a noiseless $\mathbf{g}$-LWE encryption of $\mathbf{s}_i$. Moreover, this transformation maps the uniform distribution to the uniform distribution. So, it gives a valid reduction to the problem of distinguishing (7) from the uniform distribution.

Finally, we need to change the encryption key in (7) from $\mathbf{s}_0$ to $\mathbf{s}$. This is done by subtracting $\mathbf{A}' \sum_{i \geq 1} \mathbf{s}_i$ from $\mathbf{b}'$. Again, this maps the uniform distribution to itself, and distribution (7) to the target distribution

$$\left[\mathbf{A}', \mathbf{b}' - \mathbf{A}' \sum_{i \geq 1} \mathbf{s}_i\right] = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}\left(\begin{bmatrix}\phi(\mathbf{s})\\ \mathbf{0}\end{bmatrix}\right).$$

□

To be clear, the linear circular LWE assumption does not have any implications to constructing non-leveled fully homomorphic encryption as far as we know. Nevertheless, we view the fact that Theorem 2 is true as a positive sign for the resolution of Conjecture 1. Note also that a quadratic circular encryption contains a linear circular encryption as a subset. So, $\mathsf{circLWE}$ is at least as strong as the linear LWE assumption.

3.2 Search to Decision Reduction

The search-$\mathsf{circLWE}$ assumption states that given

$$(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s})), [\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}])$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{w \times n}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \chi^w$, it is hard for probabilistic polynomial-time algorithms to recover $\mathbf{s}$ except with negligible probability. It is trivial to see that $\mathsf{circLWE}$ implies search-$\mathsf{circLWE}$. The goal of this section is to show the converse.

First, we need the following lemma which says that breaking the $\mathsf{circLWE}$ assumption for random secrets gives us a way to break $\mathsf{circLWE}$ for any secret. Such a worst-case to average-case reduction for LWE is elementary; however, it does not seem to be so for the circular LWE assumption.

Lemma 3. *There is an efficiently computable map that transforms the* circLWE *distribution* $[\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}); \chi_\sigma), \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0}; \chi)]$ *for any (fixed) secret* $\mathbf{s}$*, to the distribution* $[\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}'}(\psi(\mathbf{s}'); \chi_{\sigma'}), \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}'}(\mathbf{0}); \chi]$ *with random secret* $\mathbf{s}' = \mathbf{s} + \mathbf{r} \bmod q$ *where* $\sigma' = \sigma \cdot 2^{\omega(\log \lambda)}$.

Moreover, the map transforms $[\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}); \chi_\sigma), \mathbf{U}]$ *with a uniformly random* $\mathbf{U}$ *to* $[\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}'}(\psi(\mathbf{s}'); \chi_{\sigma'}), \mathbf{U}]$.

Proof. Recall that $\mathbf{g}\text{-}\mathsf{LWE}$ encryption is key-homomorphic, in the sense that there is an efficiently computable transformation that on input a ciphertext $[\mathbf{A}, \mathbf{b}] = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{m})$ and an arbitrary vector $\mathbf{r}$, outputs a ciphertext

$$h_{\mathbf{r}}([\mathbf{A}, \mathbf{b}]) = [\mathbf{A}, \mathbf{b} - \mathbf{A}\mathbf{r}] \tag{8}$$

in $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}+\mathbf{r}}(\mathbf{m})$. This transformation preserves the encryption error. So, if the input is a fresh $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{m})$ ciphertext, the output is also distributed as a fresh encryption of $\mathbf{m}$ under the modified key $\mathbf{s} + \mathbf{r} \bmod q$.

We also use the fact that using auxiliary information $\mathbf{P} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}))$ as an evaluation key, it is possible to perform arbitrary homomorphic computations on $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}$ ciphertexts. More specifically, there is an efficiently computable function Eval that on input any function f and ciphertext $\mathbf{C} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{m})$, outputs a ciphertext

$$\mathsf{Eval}(\mathbf{P}, f, \mathbf{C}) \in \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(f(\mathbf{m})).$$

(Such an evaluation algorithm follows from the GSW encryption scheme, e.g. as presented in Sect. 4.2.) This transformation modifies the encryption noise, but the output distribution can be made statistically close to a fresh $\mathbf{g}\text{-}\mathsf{LWE}$ encryption with larger noise parameters using the noise-flooding lemma; see Lemma 2.

Now, for any vector $\mathbf{r}$, consider the function

$$f_{\mathbf{r}}(\psi(\mathbf{x})) = \psi(\mathbf{x} + \mathbf{r}).$$

Notice that this function maps binary vectors to binary vectors, and can be represented as an arithmetic circuit. It follows that

$$\begin{aligned} h_{\mathbf{r}}(\mathsf{Eval}(\mathbf{P}, f_{\mathbf{r}}, \mathbf{C})) &= h_{\mathbf{r}}(\mathsf{Eval}(\mathbf{P}, f_{\mathbf{r}}, \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s})))) \\ &= h_{\mathbf{r}}(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(f_{\mathbf{r}}(\psi(\mathbf{s})))) \\ &= \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}+\mathbf{r}}(\psi(\mathbf{s} + \mathbf{r})). \end{aligned}$$

This allows to map the first component of the distribution $[\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}); \chi_\sigma), \mathbf{W}]$ to $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}+\mathbf{r}}(\psi(\mathbf{s} + \mathbf{r}); \chi')$ for some noise distribution χ' to be determined. For the second component $\mathbf{W} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0}; \chi)$ (or uniformly random $\mathbf{W} = \mathbf{U}$) we simply output $h_{\mathbf{r}}(\mathbf{W})$, which preserves the error distribution χ.

It remains to analyze the noise-growth χ' resulting from the homomorphic computation. First, we note that the function $f_{\mathbf{r}}$ essentially performs addition (in parallel on its coordinates) by a constant vector $\mathbf{r}$ with entries in $\mathbb{Z}_q$ followed by a single multiplication. This can be implemented with a circuit of depth

$O(\log \log q)$. By [17,31], homomorphic evaluation of $f_{\mathbf{r}}$ increases the noise magnitude from $\sigma\sqrt{\lambda}$ to at most $\sigma\sqrt{\lambda} \cdot 2^{O(\log \log q)} \cdot \mathrm{poly}(\lambda) = \sigma \cdot \mathrm{poly}(\lambda, \log q)$. Since $\log q = \mathrm{poly}(\lambda)$, the noise flooding lemma using a Gaussian with standard deviation $\sigma' = \sigma \cdot \mathrm{poly}(\lambda, \log q) \cdot 2^{\omega(\log \lambda)}$ makes this into a Gaussian with standard deviation $\sigma' = \sigma \cdot 2^{\omega(\log \lambda)}$. □

The following corollary follows immediately from Lemma 3.

Corollary 1. *If there is a probabilistic polynomial-time distinguisher for* circLWE *that works for a non-negligible fraction of secrets* $\mathbf{s} \leftarrow \mathbb{Z}_q^n$*, then there is a probabilistic polynomial-time distinguisher that, for all secrets* $\mathbf{s} \in \mathbb{Z}_q^n$*, outputs* 1 *with overwhelming probability given* $(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s})), \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0}^\ell))$ *and* 0 *with overwhelming probability given* $(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s})), \mathbf{U})$.

Finally, we state and prove our search to decision reduction for circLWE.

Theorem 3. *Let* λ *be a security parameter. If* search-circLWE *with parameters* n, q *and discrete Gaussian error distribution* χ_σ *is true, so is* circLWE *with parameters* n, q *and discrete Gaussian error distribution* $\chi_{\sigma'}$ *where* $\sigma' = \sigma \cdot 2^{\omega(\log \lambda)}$.

Proof. The proof goes along the same lines as the simple (non-sample-preserving) search to decision reduction for LWE [56, Section 4], with the only non-triviality being that the reduction needs to re-randomize the secret. This is handled by our Lemma 3. Details follow.

Let $[\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s})), \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0})]$ be the input to the search-circLWE problem. The goal is to recover $\mathbf{s}$. This is done by recovering $\mathbf{s}$ one coordinate s_i at a time, with the help of the (decisional) circLWE oracle. For any coordinate i, and guess $v \in \mathbb{Z}_q$ for the value of s_i, select some rows $[\mathbf{A}, \mathbf{b}]$ from $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0})$, choose $\mathbf{a}$ uniformly at random, and compute $[\mathbf{A}', \mathbf{b}'] = [\mathbf{A} + \mathbf{a} \cdot \mathbf{e}_i^t, \mathbf{b} + \mathbf{a} \cdot v]$. Note that $\mathbf{A}'$ is always uniformly random. Moreover, if $s_i = v$, then $\mathbf{b}'$ follows that $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\mathbf{0})$ distribution. On the other hand, if $s_i \neq v$, then $\mathbf{b}'$ is uniformly random and independent of $\mathbf{A}'$. Applying Lemma 3 to $(\mathbf{P}, [\mathbf{A}', \mathbf{b}'])$ (where $\mathbf{P} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}))$) we can randomize the secret $\mathbf{s}'$, and use the (decision) circLWE oracle to determine if $s_i = v$ was the correct guess.

Trying all possible guesses $v \pmod q$ takes time polynomial in q. For larger moduli, assume $q = \prod_i p_i$ factors into a product of small primes, and determine the value of $v \pmod{p_i}$ for each p_i separately. A reduction for arbitrary modulus q is obtained using modulus switching. □

It is natural to ask if there is a tighter reduction from search to decision, along the lines of the sample-preserving search-to-decision reduction for LWE from the work of Micciancio and Mol [49]. We conjecture that this is possible.

Conjecture 2. There is a *sample-preserving* reduction from search-circLWE to (decisional) circLWE.

3.3 Key Cliques

A natural question to ask is whether circLWE implies security of LWE when given multiple key cycles. For example, given

$$\left(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_i}(\psi(\mathbf{s}_j)) : i, j \in [k]\right)$$

for some $k > 1$, one could conjecture that the collection of $[\mathbf{A}_i, \mathbf{A}_i\mathbf{s}_i + \mathbf{e}_i]$ for all $i \in [k]$ are indistinguishable from random numbers $[\mathbf{A}_i, \mathbf{b}_i]$. We call this the k-circLWE assumption, which turns out to be equivalent to the circLWE assumption. We provide an informal statement as well as a sketch of the proof below.

Theorem 4 (Informal). circLWE *implies* k-circLWE *(i.e., Definition 5 with* $\mathsf{Pub}(\mathbf{S}) = \{\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_i}(\psi(\mathbf{s}_j))\}_{i,j}$*) for any* $k = \mathrm{poly}(n)$.

Proof (sketch). The reduction gets $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}))$, and computes

$$\left(\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_i}(\psi(\mathbf{s}_j)) : i, j \in [k]\right)$$

The reduction defines $\mathbf{s}_j$ implicitly to be $\mathbf{s} + \mathbf{r}_j$ where the reduction chooses and knows $\mathbf{r}_j \leftarrow \mathbb{Z}_q^n$. This can be done exactly as in the proof of Lemma 3: first, starting from $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}))$, compute $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s} + \mathbf{r}_j))$ using homomorphic computation, and then change the secret using the function $h_{\mathbf{r}}$ from Eq. 8 to get $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}+\mathbf{r}_i}(\psi(\mathbf{s} + \mathbf{r}_j))$. □

3.4 Other Gadgets

In this paper, we focus on the use of the power of two gadget $\mathbf{pow}(2)$, as this is the most commonly used in theoretical papers. Still, one may consider the circLWE assumption with respect to a different gadget vectors $\mathbf{h}$, i.e., given the encryption $\mathsf{Pub}_{\mathbf{h}}(\mathbf{s}) = \mathbf{h}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi_{\mathbf{h}}(\mathbf{s}))$ instead of $\mathsf{Pub}_{\mathbf{g}}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi_{\mathbf{g}}(\mathbf{s}))$. So, one may ask, how does the choice of the gadget $\mathbf{g}$ affect the circLWE assumption? Is there a way to map $\mathsf{Pub}_{\mathbf{g}}(\mathbf{s})$ to $\mathsf{Pub}_{\mathbf{h}}(\mathbf{s})$?

It is easy to map $\mathbf{C} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi_{\mathbf{g}}(\mathbf{s}))$ to $\mathbf{h}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi_{\mathbf{g}}(\mathbf{s}))$ simply by computing $\mathbf{h} \circ \mathbf{C} = (\mathbf{I} \otimes \mathbf{h}) \odot \mathbf{C}$. However, changing $\psi_{\mathbf{g}}$ into $\psi_{\mathbf{h}}$ inside the encryption seems harder. A natural approach is to use the homomorphic properties of the encryption scheme to change $\psi_{\mathbf{g}}$ into $\psi_{\mathbf{h}}$ by means of a homomorphic computation. We conjecture that this is possible (up to some parameter growth) and pose it as a problem to be addressed in future work.

4 Homomorphic Encryption Schemes

We describe a number of widely used fully homomorphic encryption (FHE) schemes along with their associated Pub functions. Our goal is not to be comprehensive, but rather to describe a set of representative examples of FHE schemes together with the circular security assumptions they rely on. For each scheme, we will describe the key generation and encryption algorithms, focusing on the

Pub function that captures the auxiliary information about the secret key that is revealed by the scheme. We do not describe the decryption function, the homomorphic operations or the bootstrapping procedure as they are not necessary for the purposes of analyzing security. For these algorithms, we point the reader to the original papers.

4.1 BV 2011, BGV 2012 and Brakerski 2012

We begin with the Brakerski-Vaikuntanathan scheme [16], the first leveled FHE scheme whose security was based on LWE. This was followed shortly after by Brakerki, Gentry and Vaikuntanathan [12] which improved one of the key techniques in [16], namely modulus switching. In a nutshell, rather than perform modulus switching once at the end of a homomorphic computation, [12] did modulus reduction at every step, resulting in a large efficiency gain. We focus here on a scheme by Brakerski [10], which further improved on this line of work by doing *implicit modulus switching*. A concrete consequence was a simpler scheme that used the same modulus throughout, whereas BV11 and BGV12 use switching keys to go between LWE ciphertexts under different moduli. Brakerski [10] also introduced a different method to homomorphically multiply ciphertexts than BV/BGV. But the methods are very similar, and, in particular, they use essentially the same evaluation keys. So, everything we say holds for either multiplication method, since it only depends on the value of the evaluation key, and not the details of the homomorphic multiplication algorithm.

Here we consider the (circular) private-key version from [10, Section 4] with the following algorithms:

- **Parameters:** The scheme uses an LWE dimension n and an integer modulus q, and the plaintext space is integers modulo the plaintext modulus $p = 2$. (This can be generalized to other $\mathbb{Z}_q$ and $\mathbb{Z}_p$ as long as p is sufficiently smaller than q.)
- **Key Generation:** The key generation algorithm Gen outputs a random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ as the private key.
- **The Pub Function:** $\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s})$ outputs

$$\mathbf{g}\text{-}\mathsf{LWE}_{q,\mathbf{s}}\big(\mathbf{g}^{-1}(\mathbf{s},1) \otimes \mathbf{g}^{-1}(\mathbf{s},1)\big) \in \mathbb{Z}_q^{((n+1)^2 w^3)\times(n+1)}$$

  where $\mathbf{g} = \mathbf{pow}(2) \in \mathbb{Z}_q^w$ is the power-of-two gadget and $w = \lfloor \log_2 q \rfloor$ is the gadget dimension.
- **Encryption:** A message $m \in \mathbb{Z}_2$ is encrypted using the trivial gadget $\mathbf{g} := (1) \in \mathbb{Z}_2$ as

$$\mathsf{Enc}_{\mathbf{s}}(m) = \mathbf{g}\text{-}\mathsf{LWE}_{q,\mathbf{s}}(m) \ .$$

Theorem 5. *Under the* $\mathsf{circLWE}[\xi, \Xi]$ *assumption, the Brakerski (private key, fully homomorphic) encryption scheme [10, Section 4] with encryption noise* ξ *and auxiliary input function* $\mathsf{Pub}_{\mathsf{B12}}[\Xi]$ *is IND-CPA-secure.*

Proof. Let $\mathcal{A}$ be an adversary that breaks the IND-CPA security of the encryption scheme with auxiliary input function $\mathsf{Pub}_{\mathsf{B12}}$, and notice that $\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s}; \Xi)$ equals precisely $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi_{\mathbf{g}}(\mathbf{s}); \Xi)$, the auxiliary information of our circular LWE assumption. We use $\mathcal{A}$ to break the circLWE problem. Recall that $\mathcal{A}$ has access to an encryption oracle $O_b(m_0, m_1)$ that on input a pair of messages m_0, m_1 returns a ciphertext $\mathsf{LWE}_{\mathbf{s}}(m_b) = (\mathbf{a}^t, \mathbf{a}^t\mathbf{s} + e + \lfloor m_b \rceil_q)$, for a fixed, randomly chosen $b \in \{0,1\}$. The goal of the adversary is to guess the bit b. We use $\mathcal{A}$ to break the circLWE assumption as follows. Let $(\mathbf{C}, [\mathbf{A}, \mathbf{b}])$ be the circLWE input distribution, where $[\mathbf{A}, \mathbf{b}]$ has a sufficiently high number of row[9] and follows either the uniform or LWE distribution with noise $\mathbf{e} \leftarrow \xi$. Pick $x \leftarrow \{0,1\}$ uniformly at random, run $\mathcal{A}(\mathbf{C})$ and every time $\mathcal{A}$ makes a call to the encryption oracle $O(m_0, m_1)$ reply with $(\mathbf{a}_i^t, b_i + \lfloor m_x \rceil_q)$ using one of the rows of $[\mathbf{A}, \mathbf{b}]$. When $\mathcal{A}$ terminates with output $y \in \{0,1\}$, the circLWE distinguisher outputs $x + y \pmod 2$. Notice that if $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, then $O(m_0, m_1) = \mathsf{LWE}_{\mathbf{s}}(m_x; \mathbf{e})$ and $\mathcal{A}$ will correctly guess the random bit x with some non-negligible advantage. On the other hand, if $\mathbf{b}$ is chosen uniformly at random, then the output of $O(m_0, m_1)$ is statistically independent of x, and $\mathcal{A}$ will output x with probability exactly $1/2$. So, the circLWE distinguisher has essentially the same running time and distinguishing advantage (up to a factor 2) as $\mathcal{A}$. □

An Optimization. It is possible to reduce the size of the switching key by letting

$$\mathsf{Pub}_{\mathsf{B12opt}}(\mathbf{s}) := \mathbf{g}\text{-}\mathsf{LWE}_{q,\mathbf{s}}(\mathbf{g}^{-1}((\mathbf{s},1) \otimes (\mathbf{s},1))) \in \mathbb{Z}_q^{((n+1)^2 w^2) \times (n+1)} ,$$

i.e., taking the binary decomposition *after* tensoring the key, rather than before. This reduces the size of the switching key by a factor of $w = \lfloor \log_2 q \rfloor$ and is an optimization employed in most subsequent papers that build on [10], e.g., the Fan-Vercauteren ring variant [24]. If $\mathbf{s} \in \{0,1\}^n$ is binary, the two Pub functions coincide. However, they are different for general secrets $\mathbf{s} \in \mathbb{Z}_q^n$.

It is clear that the one can prove the security of the resulting scheme just as in Theorem 5, using a variant of our circLWE assumption where the function $\psi(\mathbf{s}) = \mathbf{g}^{-1}(\mathbf{s},1) \otimes \mathbf{g}^{-1}(\mathbf{s},1)$ is replaced by $\psi'(\mathbf{s}) = \mathbf{g}^{-1}((\mathbf{s},1) \otimes (\mathbf{s},1))$. It is also tempting to assume that one can compute $\mathsf{Pub}_{\mathsf{B12opt}}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi'(\mathbf{s}))$ from $\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}(\psi(\mathbf{s}))$, and, thereby, establish the security of the optimized scheme under the standard circLWE assumption, possibly at the cost of using larger parameters. The idea is to express the products $u \cdot v$ in $(\mathbf{s},1) \otimes (\mathbf{s},1)$ where u, v are elements of $(\mathbf{s},1)$ as a binary circuit that takes as input the bits of $v = \sum_i v_i 2^i$ and $u = \sum_i u_i 2^i$. Then, evaluate the circuit homomorphically on the encryptions of u_i, v_i (which are available from $\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s})$) using the (leveled) homomorphic operations of the encryption scheme, producing as a result the encryption of the bits of $u \cdot v$, and concatenate them together to form $\mathsf{Pub}_{\mathsf{B12opt}}(\mathbf{s})$. The problem is that this approach produces encryptions of the bits in $\psi'(\mathbf{s})$ under the BV/BGV/B12 LWE encryption scheme with trivial gadget $\mathbf{g} = (1)$ and plaintext modulus $p = 2$,

[9] The number of rows may be fixed using a public key version of the encryption scheme.

while $\mathsf{Pub}_{\mathsf{B12opt}}(\mathbf{s})$ requires the output to be encrypted under $\mathbf{g} = \mathbf{pow}(2)$ and modulus $p = q$. So, it is unclear how to compute $\mathsf{Pub}_{\mathsf{B12opt}}(\mathbf{s})$ from $\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s})$, and at this point the optimized scheme seems to require a different (and possibly stronger) assumption than circLWE.

4.2 GSW 2013 and BV 2014

We next consider a different family of LWE-based encryption schemes that stem from the work of Gentry, Sahai and Waters [31]. Their work does not describe an explicit bootstrapping algorithm, rather it mentions that the scheme can be bootstrapped using general techniques, leading to an FHE scheme with quasi-polynomial modulus q. Brakerski and Vaikuntanathan [17] describe a new bootstrapping algorithm for the GSW scheme, leading to the first leveled FHE with a polynomial modulus q. This has been further simplified and optimized by Alperin-Sheriff and Peikert [3].

The Key-Switching Lens on GSW 2013 Ciphertexts. Focusing on the encryption scheme in [31, Section 3], we present a very different, but completely equivalent, view on GSW ciphertexts as switching keys in the Regev encryption scheme. We describe our version of GSW 2013 below, and go on to show that the ciphertexts thus generated are computationally equivalent to the ciphertexts in [31, Section 3].

- **Parameters:** The scheme uses an LWE dimension n and an integer (ciphertext) modulus q, and the plaintext space is $\mathcal{M} = \{0, 1\} \subset \mathbb{Z}_q$.
- **Key Generation:** The key generation algorithm Gen outputs a random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ as the private key.
- **Encryption:** The scheme uses the power gadget $\mathbf{g} = \mathbf{pow}(2) \in \mathbb{Z}_q^w$ in dimension $w = \lfloor \log_2 q \rfloor$.[10] A message $m \in \mathcal{M}$ is encrypted as

$$\mathsf{Enc}_{\mathbf{s}}(m) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s} \to \mathbf{s}}(m) \in \mathbb{Z}_q^{(n+1)w \times (n+1)}$$

  so ciphertexts are matrices.[11]

Before proceeding any further, we observe that these ciphertexts can be added and multiplied using the switching keys composition properties described at the end of Sect. 2.3. We leave it to the reader to verify that the properties are operationally identical to the homomorphic addition and multiplication of the GSW encryption scheme.

[10] [31] also mentions the possibility of using $\mathbf{g} = \mathbf{pow}(b)$ for some other basis b, or a CRT gadget $\mathbf{g} = \mathbf{crt}(\mathbf{p})$, though the scheme is only presented and analyzed for the specific case of $\mathbf{g} = \mathbf{pow}(2)$.

[11] The original GSW 2013 encryption scheme outputs a bit-decomposition of the mod-q matrix as the ciphertext. Here, we use an equivalent variant from [3] which outputs the mod-q matrix as-is.

We now show that these ciphertexts are exactly GSW 2013 ciphertexts. Let us first rewrite the ciphertext in the language of LWE.

$$\begin{aligned}\mathsf{Enc}_{\mathbf{s}}(m) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}\to\mathbf{s}}(m) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}\left(\begin{bmatrix}\mathbf{s}\\1\end{bmatrix} m\right) = \mathsf{LWE}_{\mathbf{s}}\left(\begin{bmatrix}\mathbf{s}\\1\end{bmatrix} m \otimes \mathbf{g}\right)\\ = \left[\mathbf{A}, -\mathbf{A}\mathbf{s} + \mathbf{e} + \begin{bmatrix} m\mathbf{s}\otimes\mathbf{g} \\ m\mathbf{g}\end{bmatrix}\right]\end{aligned}$$

Now, writing $m\mathbf{s}\otimes\mathbf{g}$ as

$$m\mathbf{s}\otimes\mathbf{g} = (m\mathbf{I}_n\otimes\mathbf{g})(\mathbf{s}\otimes 1) = (m\mathbf{I}_n\otimes\mathbf{g})\mathbf{s}\ ,$$

we can write the ciphertext as

$$\begin{aligned}\left[\mathbf{A}, -\mathbf{A}\mathbf{s} + \mathbf{e} + \begin{bmatrix} m\mathbf{s}\otimes\mathbf{g} \\ m\mathbf{g}\end{bmatrix}\right] &= \begin{bmatrix}\mathbf{A}_1, -(\mathbf{A}_1 - m\mathbf{I}_n\otimes\mathbf{g})\mathbf{s} + \mathbf{e}_1\\ \mathbf{A}_2, -\mathbf{A}_2\mathbf{s} + \mathbf{e}_2 + m\mathbf{g}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{A}'_1 + m\mathbf{I}_n\otimes\mathbf{g}, -\mathbf{A}'_1\mathbf{s} + \mathbf{e}_1\\ \mathbf{A}_2, \qquad\qquad -\mathbf{A}_2\mathbf{s} + \mathbf{e}_2 + m\mathbf{g}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{A}'_1, -\mathbf{A}'_1\mathbf{s} + \mathbf{e}_1\\ \mathbf{A}_2, -\mathbf{A}_2\mathbf{s} + \mathbf{e}_2\end{bmatrix} + m\mathbf{I}_{n+1}\otimes\mathbf{g}\\ &= [\mathbf{A}, -\mathbf{A}\mathbf{s} + \mathbf{e}] + m\mathbf{I}_{n+1}\otimes\mathbf{g}\end{aligned}$$

The latter expression is exactly a GSW 2013 ciphertext, except that (following [3]) it is written as a mod-q matrix whereas GSW 2013 further do bit-decomposition to turn it into a 0-1 matrix.

The Pub **Function for GSW.** The $\mathsf{Pub}_{\mathsf{GSW13}}$ function encrypts the bits of each coordinate of the secret key $\mathbf{s}$ under the GSW encryption algorithm:

$$\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}\to\mathbf{s}}(\mathbf{g}^{-1}(\mathbf{s})).$$

We recall that the GSW encryption scheme has message space $\{0,1\}$. So, the above expression should be interpreted as the concatenation of wn $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}\to\mathbf{s}}$ ciphertexts, each encrypting one of the elements of $\mathbf{g}^{-1}(\mathbf{s}) \in \{0,1\}^{wn}$ independently.

Theorem 6. *Under the* $\mathsf{circLWE}[\xi,\chi_\sigma]$ *assumption, the GSW (private key, fully homomorphic) encryption scheme [31, Section 3] with encryption noise* ξ *and auxiliary input function* $\mathsf{Pub}_{\mathsf{GSW13}}[\chi_{\sigma'}]$ *is IND-CPA-secure, where* $\sigma' = \sqrt{w}\cdot\sigma$.

Proof. By definition

$$\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}\to\mathbf{s}}(\mathbf{g}^{-1}(\mathbf{s})) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}\left(\mathbf{g}^{-1}(\mathbf{s})\otimes\begin{bmatrix}\mathbf{s}\\1\end{bmatrix}\right).$$

This can be generated from

$$\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s};\Xi) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}}\big(\mathbf{g}^{-1}(\mathbf{s},1)\otimes\mathbf{g}^{-1}(\mathbf{s},1)\,;\Xi\big)$$

via additive homomorphisms with noise growth $O(\sqrt{\log q})$. More specifically, one can compute

$$\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s}) = \mathbf{G} \circledcirc \mathsf{Pub}_{\mathsf{B12}}(\mathbf{s}; \Xi) \qquad \text{where} \quad \mathbf{G} = [\mathbf{I}, \mathbf{O}] \otimes (\mathbf{I} \otimes \mathbf{g}^t)$$

which, by the mixed product property of $\otimes$ and the definition of $\circledcirc$, is a $\mathbf{g}$-$\mathsf{LWE}_{\mathbf{s}}$ encryption of

$$\begin{aligned}
&\left([\mathbf{I}, \mathbf{O}] \otimes (\mathbf{I} \otimes \mathbf{g}^t)\right) \cdot \left(\mathbf{g}^{-1}\left(\begin{bmatrix}\mathbf{s}\\1\end{bmatrix}\right) \otimes \mathbf{g}^{-1}\left(\begin{bmatrix}\mathbf{s}\\1\end{bmatrix}\right)\right) \\
&= \left([\mathbf{I}, \mathbf{O}] \cdot \mathbf{g}^{-1}\left(\begin{bmatrix}\mathbf{s}\\1\end{bmatrix}\right)\right) \otimes \left((\mathbf{I} \otimes \mathbf{g}^t) \cdot \mathbf{g}^{-1}\left(\begin{bmatrix}\mathbf{s}\\1\end{bmatrix}\right)\right) \\
&= \mathbf{g}^{-1}(\mathbf{s}) \otimes \begin{bmatrix}\mathbf{s}\\1\end{bmatrix}.
\end{aligned}$$

This proves that $\mathsf{Pub}_{\mathsf{GSW13}})(\mathbf{s})$ encrypts the correct message. As for the encryption noise distribution, notice that if $\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s})$ has discrete gaussian noise distribution Ξ then $\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s})$ has noise distribution $\mathbf{g}^{-t}(\mathbf{G}) \cdot \Xi$. But, for any gadget $\mathbf{g}$, the gadget decompositions of the coordinates of the gadget vector are (trivially) the unit vectors $\mathbf{g}^{-t}(\mathbf{g}_i) = \mathbf{e}_i^t = [0, \ldots, 1, \ldots, 0] \in \{0,1\}^w$ with the 1 at position i. It follows that $\mathbf{g}^{-t}(\mathbf{G}) = [\mathbf{I} \otimes \mathbf{g}^{-t}(\mathbf{g}^t), \mathbf{O}]$ is a binary matrix with orthogonal rows of weight w. So, if Ξ follows the gaussian distribution χ_σ, then each coordinate of $\mathbf{g}^{-t}(\mathbf{G})\,\Xi$ is the sum of w independent gaussians χ_σ. So, $\mathbf{g}^{-t}(\mathbf{G})\,\Xi$ is also gaussian $\chi_{\sigma'}$ with parameter $\sigma' = \sqrt{w} \cdot \sigma$.

This is not enough to show that the IND-CPA security of GSW follows from the security of B12 because the encryption function is also different. Still, we can proceed similarly to the proof of Theorem 5 as follows. Let $\mathcal{A}$ be an adversary breaking the IND-CPA security of GSW, and let $(\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s}), \mathbf{C})$ the input for the $\mathsf{circLWE}$ problem. Here $\mathbf{C} \in \mathbb{Z}_q^{*\times(n+1)}$ is a matrix with sufficiently many rows, and it is broken into chunks $\mathbf{C}_i \in \mathbb{Z}_q^{(n+1)w\times(n+1)}$, one for each encryption query to be made by $\mathcal{A}$. The goal is to determine if $\mathbf{C}$ follows the LWE or the uniformly random distribution. First we compute $\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s})$ from $\mathsf{Pub}_{\mathsf{B12}}(\mathbf{s})$ as described above, and pick a bit $x \in \{0,1\}$ uniformly at random. Next, we run $\mathcal{A}$ on input $\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s})$, and every time $\mathcal{A}$ makes an encryption query (m_0, m_1) we reply with $\mathbf{C}_i + m_x\mathbf{I} \otimes \mathbf{g}$, where i is a counter which is incremented after every query. If $\mathbf{C}$ is follows the LWE distribution (with secret $\mathbf{s}$), $\mathbf{C}_i + m_x\mathbf{I} \otimes \mathbf{g}$ is a GSW encryption of m_x, and $\mathcal{A}$ will have some advantage in guessing the value of the bit x. On the other hand, if $\mathbf{C}_i$ is uniformly random, $\mathcal{A}$ has no information about x and will guess it with probability exactly $1/2$. So, we can determine if $\mathbf{C}$ is LWE or uniform by checking if $\mathcal{A}$ outputs x. □

4.3 AP14 and GINX16

The work of Alperin-Sheriff and Peikert [3] builds on GSW 2013 and BV 2014, but uses a different bootstrapping procedure requiring a different encoding of the secret key. Several encoding methods are described in [3]. In the simplest

(but least efficient) method each coordinate $s_i \in \mathbb{Z}_q$ of the secret key is encoded as a permutation matrix $\Pi_i \in \{0,1\}^{q\times q}$ such that $\Pi_i \mathbf{x}$ rotates the vector $\mathbf{x}$ by s_i positions. Then, the entries of Π_i (for each i) are encrypted using the GSW encryption function. Then a number of optimizations are considered. First, since each row of Π_i is a rotation of the previous row, there is no need to encrypt all $q \times q$ entries: it is enough to provide encryptions of the first row π_i, producing just q ciphertexts for each i. (For the other rows one can use rotations of those q ciphertexts.) Even more substantial savings can be obtained when $q = \prod_j q_j$ is a product of small primes. Then, one can use the isomorphism between $\mathbb{Z}_q$ and $\prod_j \mathbb{Z}_{q_j}$, map the secret key to a collection of values $s_{i,j} = s_i \pmod{q_j}$, and then encode each $s_{i,j} \in \mathbb{Z}_{q_j}$ as before as (the first row of) a permutation matrix in dimension q_j. Our proof and security analysis holds for all different variants of the encoding functions, as it operates on each vector individually. For concreteness, we consider the direct encoding of $x \in \mathbb{Z}_q$ as a q-dimensional binary vector. But the proof is immediately adapted to the case of composite $q = \prod_j q_j$, breaking $\mathbb{Z}_q$ into the product of smaller cycles.

In summary, the $\mathsf{Pub}_{\mathsf{AP14}}$ function encrypts a one-hot encoding of each coordinate $s_i \in \mathbb{Z}_q$ (or $s_{i,j} \in \mathbb{Z}_{q_j}$) of the secret key. For a number $x \in \mathbb{Z}_q$, let $e_x \in \{0,1\}^q$ denote the vector with 1 in the x^{th} coordinate and 0 everywhere else. Extending the notation to vectors, for $\mathbf{s} \in \mathbb{Z}_q^n$, let $e_{\mathbf{s}} \in \{0,1\}^{nq}$ denote the vertical concatenation of the e_{s_i} for all i.

$$\mathsf{Pub}_{\mathsf{AP14}}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}\to\mathbf{s}}(e_{\mathbf{s}})$$

Theorem 7. *Under the* $\mathsf{circLWE}[\xi,\chi_\sigma]$ *assumption, the AP14 (private key, fully homomorphic) encryption scheme [3, Section 5] with encryption noise* ξ *and auxiliary input function* $\mathsf{Pub}_{\mathsf{AP14}}[\Xi]$ *is IND-CPA-secure for some efficiently samplable distribution* Ξ *with subgaussian parameter* $|\Xi| \le w^2\sqrt{n+1}\cdot\sigma$.

Proof. Since AP14 and GSW use the same encryption function, it is enough to show how to generate $\mathsf{Pub}_{\mathsf{AP14}}(\mathbf{s})$ from $\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s})$. Then, security follows from the proof of Theorem 6. Recall that $\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}\to\mathbf{s}}(\mathbf{x}^t)$ where $\mathbf{x}^t = \mathbf{g}^{-t}(\mathbf{s})$ is the bit decomposition of the secret key $\mathbf{s}$. Moreover, Theorem 6 shows that (starting from the $\mathsf{circLWE}[\xi,\chi_\sigma]$ auxiliary information) these ciphertexts can be computed with gaussian encryption noise $\chi_{\sigma'}$ for $\sigma' = \sqrt{w}\sigma$. Consider the function $\phi(x_1,\ldots,x_w)$ that takes the bits $x_i \in \{0,1\}$ of a number $x \in \mathbb{Z}_q$ and outputs its one-hot encoding e_x. For any $y = \sum_i 2^i y_i$ (with $y_i \in \{0,1\}$), the y^{th} element of the one-hot encoding $e_x[y]$ can be written as $e_x[y] = \prod_{i=0}^{w} z_i$ where

$$\begin{aligned} z_i &= (y_i \cdot x_i + (1-y_i)\cdot(1-x_i)) \\ &= ((2y_i-1)x_i + 1 - y_i) = \begin{cases} 1 \text{ if } x_i = y_i \\ 0 \text{ otherwise} \end{cases} \end{aligned}$$

Notice that $(2y_i - 1) = \pm 1$, and it is a known constant. So, the product $(2y_i - 1)x_i$ can be evaluated homomorphically on the encryption of x_i (provided by $\mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s})$) while preserving the error distribution $\pm\chi_{\sigma'} = \chi_{\sigma'}$. Moreover,

all factors z_i in the product evaluate to either 0 or 1. So, the product can be evaluated using AP14 homomorphic multiplications. More specifically, if $\mathbf{C}_i = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}\to\mathbf{s}}(z_i;\chi_{\sigma'})$ are the encryptions of the secret bits z_i computed as above, then an encryption of $e_x[y] = \prod_{i=1}^{w} z_i$ can be computed as the product

$$\mathbf{C}_1 \odot \mathbf{C}_2 \odot \cdots \odot \mathbf{C}_w.$$

By Theorem 1, evaluating this products left-to-right gives an encryption of $e_x[y]$ with an error which is the sum of at most w terms for the form $\mathbf{g}^{-t}(\mathbf{C}'_i) \cdot \chi_{\sigma'}$, where $\mathbf{C}'_i = \mathbf{C}_1 \odot \cdots \odot \mathbf{C}_{i-1}$ are the ciphertexts corresponding to the intermediate partial products. Since the rows of $\mathbf{g}^{-t}(\mathbf{C}'_i) \in \{0,1\}^{(n+1)w\times(n+1)w}$ have norm at most $\sqrt{w(n+1)}$, each error component has gaussian distribution of parameter (at most) $\sqrt{w(n+1)}\sigma' = w\sqrt{n+1}\sigma$. Adding up the errors for all (at most w) terms, we see that $e_x[y]$ is encrypted with gaussian noise $\chi_{\sigma''}$ of parameter at most $\sigma'' \leq w^2\sqrt{n+1}\sigma$. We remark that while each $e_x[y]$ is encrypted using gaussian noise, these error distributions (for different indexes y) are not totally independent because they are obtained by taking different linear combinations of the same $\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}\to\mathbf{s}}(\mathbf{x}^t;\chi_{\sigma'})$. Independence (and a slightly better bound) can be achieved by evaluating the $\odot$ products using the subgaussian decomposition technique of [3]. □

We remark that the fact that auxiliary information noise Ξ in Theorem 7 is not a Gaussian is just an artifact of the proof, and using discrete gaussian noise $\chi_{\sigma''}$ with the same parameter $\sigma'' = w^2\sigma\sqrt{n+1}$ is only expected to improve the security of the scheme. Alternatively, a formal statement can be obtained by using noise flooding to map the error distribution Ξ resulting from the homomorphic evaluation process into a discrete gaussian, at the cost of substantially increasing the noise level.

Gama et al. [25, Section 7] give yet another bootstrapping procedure,[12] similar to AP14, but offering some advantages when the secret key has binary entries $\mathbf{s} \in \{0,1\}^n$. Arbitrary keys are mapped to binary ones by taking their binary decomposition. (See [51] for a comparison of the two methods and their relation to ring versions of the same schemes [21,23].) Since the bootstrapping key consists of the GSW encryption of the bits of the secret key $\mathbf{s}$, the Pub function is precisely the same as in the GSW scheme

$$\mathsf{Pub}_{\mathsf{GINX16}}(\mathbf{s}) = \mathsf{Pub}_{\mathsf{GSW13}}(\mathbf{s}).$$

So, it immediately follows from Theorem 6 that the scheme is secure under the $\mathsf{circLWE}$ assumption.

Theorem 8. *Under the* $\mathsf{circLWE}[\xi,\chi_\sigma]$ *assumption, the (private key, fully homomorphic) GINX encryption scheme [25, Section 7] with encryption noise* ξ *and auxiliary input function* $\mathsf{Pub}_{\mathsf{GINX16}}[\chi_{\sqrt{w}\sigma}]$. *is IND-CPA-secure.*

[12] Gama et al. also present an abstract generalization of GSW, and present the scheme using a rather nonstandard notation. But the scheme is essentially the same as GSW. So, for simplicity we present it using standard LWE notation.

4.4 HAO15

Hiromasa, Abe and Okamoto [37, Section 3] proposed a homomorphic encryption scheme called MatrixGSW, a variant of GSW 2013 which encrypts matrices directly. The private key version of the scheme is defined as follows:

- **Parameters:** The scheme uses an LWE dimension n, an integer ciphertext modulus q, and the an integer k which defines the message space $\mathcal{M} = \{0,1\}^{k\times k} \subset \mathbb{Z}_q^{k\times k}$.
- **Key Generation:** The secret key generation algorithm Gen outputs a random $n \times k$ integer matrix with small entries $\mathbf{S} \leftarrow \mathbb{Z}_q^{n\times k}$.
- **Encryption:** The encryption of a message $\mathbf{M}$ is

$$\mathsf{Enc}_{\mathbf{S}}(\mathbf{M}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}\to\mathbf{S}}(\mathbf{M})$$

  where $\mathbf{g} = \mathbf{pow}(2)$.

Interestingly, even without bootstrapping, the IND-CPA security of MatrixGSW does not seem to follow from the standard LWE assumption.[13] The security of (the public key version of) the scheme is claimed [37, Lemma 4] under an unspecified "circular security" assumption. Here we provide a sketch of the proof that the private-key encryption scheme is secure under the key clique assumption from Sect. 3.3.

Theorem 9 (Informal). *Under the key clique assumption from Sect. 3.3, the HAO15 encryption scheme [37] is IND-CPA secure.*

Proof (Sketch). Expanding the definition of $\mathsf{Enc}_{\mathbf{S}}$ we see that

$$\begin{aligned}\mathsf{Enc}_{\mathbf{S}}(\mathbf{M}) = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}\to\mathbf{S}}(\mathbf{M}) &= \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}}\left(\begin{bmatrix}\mathbf{S}\\ \mathbf{I}\end{bmatrix}\mathbf{M}\right)\\ &= \begin{bmatrix}\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}}(\mathbf{SM})\\ \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}}(\mathbf{M})\end{bmatrix}\end{aligned}$$

The bottom part is just a $\mathbf{g}$-LWE encryption of the matrix $\mathbf{M}$ under $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_k]$, and its security follows from the standard LWE assumption using a standard hybrid argument. For the top part, we show how to compute $\mathbf{C} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{S}}(\mathbf{SM})$ from the key clique $\mathbf{Q}_{i,j} = \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_j}(\mathbf{s}_i)$. Let $m_{i,j}$ be the entries of the message $\mathbf{M}$. Then, the jth column of $\mathbf{C}$ can be written as

$$\mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_j}\left(\sum_i \mathbf{s}_i \cdot m_{i,j}\right) = \sum_i m_{i,j} \overset{\mathbf{g}}{\circledcirc} \mathbf{g}\text{-}\mathsf{LWE}_{\mathbf{s}_j}(\mathbf{s}_i) = \sum_i m_{i,j} \overset{\mathbf{g}}{\circledcirc} \mathbf{Q}_{i,j}.$$

[13] [37] claims that the private key (but not the public key) version of the encryption scheme is secure under the standard decisional LWE assumption, but without giving a proof. However, the claim is probably incorrect as private key homomorphic encryption schemes can be generically transformed into public key homomorphic schemes without additional security assumptions.

In fact, if $\mathbf{M}$ is a matrix with binary entries, then one can use a simple product $m_{i,j} \cdot \mathbf{Q}_{i,j}$ instead of $\overset{\mathbf{g}}{\circledcirc}$. This expression produces an encryption of $\mathbf{SM}$, but with a different error distribution than the standard encryption function. Still, one can guarantee IND-CPA security under the key clique assumption by adding a random encryption of 0 with flooding noise, and using the noise flooding lemma (Lemma 2). □

Combining this with Theorem 4 we get security under our circLWE assumption.

Corollary 2. *Under the* circLWE *assumption, the HAO15 encryption scheme [37] is IND-CPA secure.*

Acknowledgement. DM was supported by an Intel Cryptographic Frontiers grant, SAIT Global Research Cluster and NSF award CNS-1936703. VV was supported by a Simons Investigator Award, NSF CNS-2154149, DARPA under Agreement No. HR00112020023, a grant from the MIT-IBM Watson AI and a Thornton Family Faculty Research Innovation Fellowship from MIT. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

References

1. Abadi, M., Rogaway, P.: Reconciling two views of cryptography (the computational soundness of formal encryption). J. Cryptol. **20**(3), 395 (2007). https://doi.org/10.1007/s00145-007-0203-0
2. Ajtai, M.: Generating hard instances of lattice problems. In: Symposium on Theory of Computing - STOC 1996, pp. 99–108. ACM (1996). https://doi.org/10.1145/237814.237838
3. Alperin-Sheriff, J., Peikert, C.: Faster bootstrapping with polynomial error. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 297–314. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_17
4. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_35
5. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_29
6. Barak, B., Haitner, I., Hofheinz, D., Ishai, Y.: Bounded key-dependent message security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 423–444. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_22
7. Bishop, A., Hohenberger, S., Waters, B.: New circular security counterexamples from decision linear and learning with errors. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 776–800. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_32

8. Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of key-dependent messages. In: Nyberg, K., Heys, H. (eds.) SAC 2002. LNCS, vol. 2595, pp. 62–75. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36492-7_6
9. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_7
10. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_50
11. Brakerski, Z., Döttling, N.: Hardness of LWE on general entropic distributions. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12106, pp. 551–575. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_19
12. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans. Comput. Theory **6**(3), 13:1–13:36 (2014). https://doi.org/10.1145/2633600. (Prelim. version in ITCS 2012)
13. Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_1
14. Brakerski, Z., Goldwasser, S., Kalai, Y.T.: Black-box circular-secure encryption beyond affine functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 201–218. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_13
15. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Symposium on Theory of Computing - STOC 2013, pp. 575–584. ACM (2013). https://doi.org/10.1145/2488608.2488680
16. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. SIAM J. Comput. **43**(2), 831–871 (2014). https://doi.org/10.1137/120868669. (Prelim. version in FOCS 2011)
17. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: Innovations in Theoretical Computer Science - ITCS 2014, pp. 1–12. ACM (2014). https://doi.org/10.1145/2554797.2554799
18. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_7
19. Canetti, R., Lin, H., Tessaro, S., Vaikuntanathan, V.: Obfuscation of probabilistic circuits and applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 468–497. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_19
20. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 409–437. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_15
21. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: TFHE: fast fully homomorphic encryption over the torus. J. Cryptol. **33**(1), 34–91 (2020). https://doi.org/10.1007/s00145-019-09319-x
22. TU Darmstadt Lattice Challenge. https://www.latticechallenge.org/

23. Ducas, L., Micciancio, D.: FHEW: bootstrapping homomorphic encryption in less than a second. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 617–640. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_24
24. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, p. 144 (2012). http://eprint.iacr.org/2012/144
25. Gama, N., Izabachène, M., Nguyen, P.Q., Xie, X.: Structural lattice reduction: generalized worst-case to average-case reductions and homomorphic cryptosystems. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 528–558. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_19
26. Gay, R., Pass, R.: Indistinguishability obfuscation from circular security. In: Symposium on Theory of Computing - STOC 2021, pp. 736–749. ACM (2021). https://doi.org/10.1145/3406325.3451070
27. Genise, N., Micciancio, D.: Faster Gaussian sampling for trapdoor lattices with arbitrary modulus. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 174–203. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_7
28. Genise, N., Micciancio, D., Polyakov, Y.: Building an efficient lattice gadget toolkit: Subgaussian sampling and more. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 655–684. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_23
29. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Symposium on Theory of Computing - STOC 2009, pp. 169–178. ACM (2009). https://doi.org/10.1145/1536414.1536440
30. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Symposium on Theory of Computing - STOC 2008, pp. 197–206. ACM (2008). https://doi.org/10.1145/1374376.1374407
31. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_5
32. Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: Innovations in Computer Science - ICS 2010, pp. 230–240. Tsinghua University Press (2010). http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/19.html
33. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984). https://doi.org/10.1016/0022-0000(84)90070-9
34. Goyal, R., Koppula, V., Waters, B.: Separating IND-CPA and circular security for unbounded length key cycles. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10174, pp. 232–246. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54365-8_10
35. Goyal, R., Koppula, V., Waters, B.: Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10211, pp. 528–557. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_18
36. Hiromasa, R., Abe, M., Okamoto, T.: Packing messages and optimizing bootstrapping in GSW-FHE. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 699–715. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_31
37. Hiromasa, R., Abe, M., Okamoto, T.: Packing messages and optimizing bootstrapping in GSW-FHE. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **99-A**(1), 73–82 (2016). https://doi.org/10.1587/transfun.E99.A.73

38. Hopkins, S., Jain, A., Lin, H.: Counterexamples to new circular security assumptions underlying iO. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12826, pp. 673–700. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84245-1_23
39. Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. In: Symposium on Theory of Computing - STOC 2021, pp. 60–73. ACM (2021). https://doi.org/10.1145/3406325.3451093
40. Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC0. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology - EUROCRYPT 2022. LNCS, vol. 13275, pp. 670–699. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06944-4_23
41. Kaliski, B.: Announcement of RSA factoring challenge (1991). https://groups.google.com/u/1/g/sci.crypt/c/AA7M9qWWx3w/m/EkrsR69CDqIJ
42. Koppula, V., Ramchen, K., Waters, B.: Separations in circular security for arbitrary length key cycles. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 378–400. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_15
43. Koppula, V., Waters, B.: Circular security separations for arbitrary length cycles from LWE. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 681–700. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_24
44. Li, B., Micciancio, D.: On the security of homomorphic encryption on approximate numbers. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 648–677. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_23
45. Li, B., Micciancio, D., Schultz, M., Sorrell, J.: Securing approximate homomorphic encryption using differential privacy. In: Dodis, Y., Shrimpton, T. (eds.) Advanced in Cryptology - CRYPTO 2022. LNCS, vol. 13507, pp. 560–589. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15802-5_20
46. Micciancio, D.: Improving lattice based cryptosystems using the Hermite normal form. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 126–145. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44670-2_11
47. Micciancio, D.: Computational soundness, co-induction, and encryption cycles. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 362–380. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_19
48. Micciancio, D.: On the hardness of learning with errors with binary secrets. Theory Comput. **14**(1), 1–17 (2018). https://doi.org/10.4086/toc.2018.v014a013
49. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_26
50. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
51. Micciancio, D., Polyakov, Y.: Bootstrapping in FHEW-like cryptosystems. In: Workshop on Encrypted Computing & Applied Homomorphic Cryptography - WAHC 2021, pp. 17–28. ACM (2021). https://doi.org/10.1145/3474366.3486924
52. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007). https://doi.org/10.1137/S0097539705447360

53. Micciancio, D., Walter, M.: On the bit security of cryptographic primitives. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 3–28. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_1
54. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: Symposium on Theory of Computing - STOC 2009, pp. 333–342. ACM (2009). https://doi.org/10.1145/1536414.1536461
55. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_31
56. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 34:1–34:40 (2009). https://doi.org/10.1145/1568318.1568324
57. Rothblum, R.: Homomorphic encryption: from private-key to public-key. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 219–234. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_14
58. Wee, H., Wichs, D.: Candidate obfuscation via oblivious LWE sampling. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12698, pp. 127–156. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77883-5_5